



## Topology Configuration Variables

[Topology Global-Function API](#)

---

Topology configuration variables include three subsets:

<a href="#">Cleanup variables</a>	Properties for cleanup models.
<a href="#">Cleanup action variables</a>	Properties for cleanup actions.
<a href="#">Topology variables</a>	Properties for topologies.

To allocate a set of configuration variables, regardless of which subset you will be using, use [tpm\\_varalloc](#), which returns an `ade_id` for the set that it creates.

To free a set of configuration variables, use [tpm\\_varfree](#).

To get the value of a configuration variable, use [tpm\\_varget](#).

To set the value of a configuration variable, use [tpm\\_varset](#).

To get the properties of a given cleanup action, first allocate a new set of variables, and then use [tpm\\_cleanactionlistgetat](#).

To get the properties of a given topology, first allocate a new set of variables, and then use [tpm\\_infobuildvar](#).

To list all the values of a given set of variables, use [tpm\\_varlist](#).

## Data Extension: Class-Based API

The API for data extension covers the following areas of functionality:

- Queries, including Query Libraries and Range Tables
- Drawing Sets, including Drawings and Drive Aliases
- Object Data
- Operations on Queried Objects, including Property Alteration
- Application Options
- User Management
- Coordinate Transformation

<a href="#">Object Model</a>	Containment relationships.
<a href="#">Inheritance</a>	Inheritance relationships.
<a href="#">Classes</a>	Classes sorted by functional group.
<a href="#">Functions</a>	
<a href="#">Types</a>	
<a href="#">Enumerations</a>	
<a href="#">Error Codes and Error Types</a>	

# Data Extension Object Model

[Inheritance](#) [Legend](#)

Containment relationships.

[AcMapSession \(Model Root\)](#) [AcMapProject](#)  
[AcMapODContainer](#)  
[AcMapQuery](#)

# Data Extension Inheritance

[Object Model](#)

Inheritance relationships.

[Query Classes](#) [Boundary Classes](#)

# Data Extension Classes

## Session Classes

[acmapsession](#) [acmapsessionoptionsreactor](#)  
[acmapsessionreactor](#)

## Project Classes

[acmapproject](#)  
[acmapprojectiterator](#)  
[acmapprojectoptionsreactor](#)

## Drive-Alias Classes

[acmapaliases](#)  
[acmapaliasesreactor](#)  
[acmapdrivealias](#)

## Drawing Set Classes

[acmapattacheddrawing](#)  
[acmapdrawingset](#)  
[acmapdrawingsetreactor](#)

## Drawing Object ID (Entity ID) Class

[acmapobjectid](#)

## Expression Class

[acmapexpression](#)

## Object Data Classes

[acmapodcolumndefinition](#)  
[acmapodcontainer](#)  
[acmapodrecorditerator](#)  
[acmapodtable](#)  
[acmapodtabledefinition](#)  
[acmapodtablerecord](#)  
[acmapvalue](#)

## Query Classes

[acmapquery](#)  
[acmapqueryattribute](#)  
[acmapquerybranch](#)  
[acmapquerycategory](#)  
[acmapqueryunit](#)  
[acmapreporttemplate](#)

[acmaptemplateline](#)  
[acmapsaveset](#)

### **Query Library Classes**

[acmapquerylibrary](#)  
[acmapquerylibraryreactor](#)

### **Query Condition Classes**

[acmapquerycondition](#)  
[acmapdatacondition](#)  
[acmaplocationcondition](#)  
[acmappropertycondition](#)  
[acmapsqlcondition](#)

### **Query Boundary Classes**

[acmapallboundary](#)  
[acmapbufferfenceboundary](#)  
[acmapbufferpolylineboundary](#)  
[acmapcircleboundary](#)  
[acmapclosedpolylineboundary](#)  
[acmapfenceboundary](#)  
[acmaplocationboundary](#)  
[acmapappointboundary](#)  
[acmappolygonboundary](#)  
[acmappolylineboundary](#)  
[acmapwindowboundary](#)

### **Property Alteration Classes**

[acmappropertyalteration](#)  
[acmappropertyalterationdefinition](#)  
[acmaphatchalteration](#)  
[acmaptextalteration](#)

### **Range Table Classes**

[acmaprangelibrary](#)  
[acmaprangeline](#)  
[acmaprangetable](#)

### **Error Classes**

[acmaperrorentry](#)  
[acmaperrorparameter](#)  
[acmaperrorstack](#)

### **Utility Classes**

[acmapobjarray](#)  
[acmapobjptrarray](#)  
[acmapstringarray](#)

# Data Extension Functions



[AcMapGetSession](#)

# Data Extension Types

## **AcMapId**

typedef unsigned long AcMapId;

**File** MapConstants.h

## **AcMapObjectIdArray**

typedef AcMapObjArray<AcMapObjectId> AcMapObjectIdArray;

**File** MapArxApi.h

## **AcMapOperandArray**

typedef AcMapObjPtrArray<AcMapQueryUnit> AcMapOperandArray;

**File** MapQuery.h

## **AcMapVertexArray**

typedef AcMapObjArray<AcMapPolylineVertex> AcMapVertexArray;

**File** MapBoundary.h

# Data Extension Enumerations

Enumeration members of struct AcMap.

[EAdeDwgStatus](#) [EAdeDwgUpdateStatus](#)

[EAlterationType](#)

[EClassId](#)

[EConditionOperator](#)

[EDataQueryType](#)

[EDataType](#)

[EErrCode](#)

[EErrType](#)

[EJoinOperator](#)

[ELocationType](#)

[EOpenMode](#)

[EPrefType](#)

[EPreviewDefinitionsFrom](#)

[EProjectOptionType](#)

[EPropertyType](#)

[EQueryDialogOptions](#)

[EQueryType](#)

[ERangeOperator](#)

[ESaveQueryOptions](#)

[ETableType](#)

[EUserRights](#)

[SaveSetObjectType](#)

# Data Extension Error Codes and Error Types

The error enumerations are members of struct [AcMap](#).

[EErrCode](#) [EErrType](#)

## Data Extension Global-Function API (Deprecated)

Except for [coordinate transformation](#) functions, the global function API for data extension is deprecated in favor of the [class-based API](#).

The API for data extension covers the following areas of functionality:

- Queries, including Query Libraries and Range Tables
- Drawing Sets, including Drawings and Drive Aliases
- Object Data
- Operations on Queried Objects, including Property Alteration
- Application Options
- User Management
- Coordinate Transformation

<a href="#">Data Extension Function Catalog</a>	Data extension functions sorted by name.
<a href="#">Data Extension Function Synopsis</a>	Data extension functions sorted by functional group.
<a href="#">Data Extension Types</a>	
<a href="#">Data Extension Constants</a>	



# Data Extension Function Catalog

[Data Extension Global-Function API](#)

---

Data Extension functions sorted by name.

**Note** Except for [coordinate transformation](#) functions, the global function API is for data extension is deprecated in favor of the [class-based API](#).

[alias](#) | [altp](#) | [ds](#) | [dwg](#) | [edit](#) | [ent](#) | [err](#) | [expr](#) | [key](#) | [od](#)

[os](#) | [pref](#) | [proj](#) | [ql](#) | [qry](#) | [rt](#) | [save](#) | [sql](#) | [ss](#) | [user](#) | [ver](#)

[ade\\_aliasadd](#) [ade\\_aliasdelete](#)

[ade\\_aliasgetlist](#)

[ade\\_aliasupdate](#)

[ade\\_altpclear](#)

[ade\\_altpdefine](#)

[ade\\_altpdelprop](#)

[ade\\_altpgetprop](#)

[ade\\_altplist](#)

[ade\\_altpsetprop](#)

[ade\\_dsattach](#)

[ade\\_dsdetach](#)

[ade\\_dsisnested](#)

[ade\\_dslist](#)

[ade\\_dsproplist](#)

[ade\\_dwgactivate](#)

[ade\\_dwgactualpath](#)

[ade\\_dwgaliaspath](#)

[ade\\_dwgattriblist](#)

[ade\\_dwgdeactivate](#)

[ade\\_dwggetid](#)

[ade\\_dwggetsetting](#)

[ade\\_dwghaslocks](#)

[ade\\_dwgindex](#)

[ade\\_dwgindexdef](#)

[ade\\_dwgisactive](#)

[ade\\_dwgistoplevel](#)  
[ade\\_dwgproplist](#)  
[ade\\_dwgquickview](#)  
[ade\\_dwgselectdlg](#)  
[ade\\_dwgsetof](#)  
[ade\\_dwgsetsetting](#)  
[ade\\_dwgunlock](#)  
[ade\\_dwgzoomextents](#)

[ade\\_editdefcen](#)  
[ade\\_editlockederased](#)  
[ade\\_editislocked](#)  
[ade\\_editlocked](#)  
[ade\\_editlockobjs](#)  
[ade\\_editnew](#)  
[ade\\_editunlockobjs](#)

[ade\\_entsetlocation](#)

[ade\\_errclear](#)  
[ade\\_errcode](#)  
[ade\\_errgetlevel](#)  
[ade\\_errmsg](#)  
[ade\\_errpush](#)  
[ade\\_errpushstatement](#)  
[ade\\_errqty](#)  
[ade\\_errsetlevel](#)  
[ade\\_errshowdlg](#)  
[ade\\_errstatement](#)  
[ade\\_errtype](#)

[ade\\_expreval](#)

[ade\\_keycolumnlist](#)

[ade\\_odaddfield](#)  
[ade\\_odaddrecord](#)  
[ade\\_odattachrecord](#)  
[ade\\_oddefinetab](#)  
[ade\\_oddeletefield](#)  
[ade\\_oddeletetab](#)  
[ade\\_oddelrecord](#)  
[ade\\_odfreerec](#)  
[ade\\_odgetfield](#)  
[ade\\_odgetrecfield](#)  
[ade\\_odgetrecord](#)

[ade\\_odgettables](#)

[ade\\_odmodifyfield](#)

[ade\\_odmodifytab](#)

[ade\\_odnewrecord](#)

[ade\\_odpresetfield](#)

[ade\\_odrecordqty](#)

[ade\\_odsetfield](#)

[ade\\_odtabledefn](#)

[ade\\_odtablelist](#)

[ade\\_osfexpand](#)

[ade\\_prefgetval](#)

[ade\\_prefsetval](#)

[ade\\_projgetctgname](#)

[ade\\_projgetinfo](#)

[ade\\_projgetwscode](#)

[ade\\_projlistcrdsysts](#)

[ade\\_projlistctgy](#)

[ade\\_projptbackward](#)

[ade\\_projptforward](#)

[ade\\_projsetdest](#)

[ade\\_projsetsrc](#)

[ade\\_projsetwscode](#)

[ade\\_qldelctgy](#)

[ade\\_qldelquery](#)

[ade\\_qlgetctgyinfo](#)

[ade\\_qlgetqryinfo](#)

[ade\\_qllistctgy](#)

[ade\\_qlloadqry](#)

[ade\\_qlqrygetid](#)

[ade\\_qlsetctgname](#)

[ade\\_qlsetquery](#)

[ade\\_qryclear](#)

[ade\\_qrydefine](#)

[ade\\_qryexecute](#)

[ade\\_qrygetcond](#)

[ade\\_qrygetdwgandhandle](#)

[ade\\_qrygetentlist](#)

[ade\\_qrygetreprtransform](#)

[ade\\_qrygroup](#)

[ade\\_qryhandent](#)

[ade\\_qrylist](#)  
[ade\\_qrysave](#)  
[ade\\_qrysetaltprop](#)  
[ade\\_qrysetcond](#)  
[ade\\_qrysetreprtransform](#)  
[ade\\_qrysettype](#)  
[ade\\_qryungroup](#)  
  
[ade\\_rtdefrange](#)  
[ade\\_rtdetable](#)  
[ade\\_rtgetid](#)  
[ade\\_rtgetprop](#)  
[ade\\_rtolist](#)  
  
[ade\\_saveobjs](#)  
[ade\\_savetodwg](#)  
  
[ade\\_sqlgetenvstring](#)  
  
[ade\\_ssfree](#)  
  
[ade\\_userget](#)  
[ade\\_usergetrights](#)  
[ade\\_userlist](#)  
[ade\\_userset](#)  
[ade\\_usersetrights](#)  
  
[ade\\_version](#)



# Data Extension Function Synopsis

[Data Extension Global-Function API](#)

---

Data Extension functions sorted by functional group.

**Note** Except for [coordinate transformation](#) functions, the global function API for data extension is deprecated in favor of the [class-based API](#).

[Coordinate Transformation Functions](#) [Drawing Functions](#)

[Drawing Set Functions](#)

[Drive Alias Functions](#)

[Error Message Functions](#)

[Expression Evaluation Function](#)

[Object Data Functions](#)

[Object Editing Functions](#)

[Object Saving Functions](#)

[Option Functions](#)

[Property Alteration Functions](#)

[Query Functions](#)

[Query Library Functions](#)

[Range Table Functions](#)

[SQL Environment Functions](#)

[User Security Functions](#)

[Other Functions](#)



## Data Extension Types

[Data Extension Global-Function API](#)

---

### **ade\_boolean**

```
typedef unsigned short ade_boolean;
```

### **ade\_id**

```
typedef double ade_id;
```

### **File**

AdeAds.h



## Data Extension Constants

[Data Extension Global-Function API](#)

---

```
ADEMEMERROR -1001
ADEINVEERROR -1002
ADEERROR -1003
ADE_REALFAIL -1.0
ADE_NULLID 0
ADE_TRUE 1
ADE_FALSE 0
```

### File

AdeAds.h

Annotation

▣ Namespaces

[AcMapAnnotationManager](#) Manages annotation features.

## Links

[Annotation](#), [Structures](#)

AcMapAnnotationManager Namespace

[Annotation](#) | [Structures](#)

Manages annotation features. For more information, search for *annotation* in AutoCAD Map Help.

### ☐ Enumerations



[eAnnotationExpressionFields](#) Enumerates annotation expression fields.



[eInsertPrecedenceMode](#) Enumerates annotation insert precedence modes.

### ☐ Functions

[AnnotationBlockReferenceAssociatedObjectId](#) Returns the object ID of the entity that an annotation reference references.

[AnnotationTemplateBlockDefinitionId](#) Returns an annotation template's block table record.

[AnnotationTemplateExists](#) Determines whether an annotation template exists.

[AnnotationTemplateReferencedObjIds](#) Lists all annotation references of an annotation template in the current drawing.

[CreateAnnotationTemplate](#) Creates an annotation template.

[CreateAnnotationText](#) Creates a new annotation text object in a template.

[DeleteAnnotationTemplate](#) Deletes an annotation template.

[GetExpressionString](#) Retrieves an expression that is stored with an annotation template, annotation reference, or annotation text object.

[GetTemplateColor](#) Retrieves the block color property of an annotation template. See also [SetTemplateColor\(\)](#).

[GetTemplateLayer](#)

Retrieves the block layer property of an annotation template. See also [SetTemplateLayer\(\)](#).

[GetTemplateLinetype](#)

Retrieves the block linetype property of an annotation template. See also [SetTemplateLinetype\(\)](#).

[GetTemplateLineWeight](#)

Retrieves the block linewidth property of an annotation template. See also [SetTemplateLineWeight\(\)](#).

[GetTemplateNames](#)

Lists all the annotation templates defined in the current drawing.

[GetTemplateRotation](#)

Retrieves the block rotation property of an annotation template. See also [SetTemplateRotation\(\)](#).

[GetTemplateScaleFactor](#)

Retrieves the block scale factor property of an annotation template. See also [SetTemplateScaleFactor\(\)](#).

[InsertAnnotationReference](#)

Attaches an annotation reference to an associated entity.

[InsertAnnotationReference](#)

Attaches an annotation reference to an associated entity.

[InsertAnnotationReference](#)

Attaches an annotation reference to an associated entity.

[InsertAnnotationReference](#)

Attaches an annotation reference to an associated entity.

[InsertAnnotationReferences](#)

Attaches an annotation reference to one or more

[InsertAnnotationReferences](#)

[IsAnnotationBlockReference](#)

[IsAnnotationTemplate](#)

[IsAnnotationTemplateReferenced](#)

[IsAnnotationText](#)

[RefreshAnnotationReferences](#)

[SetAnnotationInsertPrecedenceMode](#)

[SetExpressionString](#)

[SetExpressionString](#)

[SetTemplateColor](#)

[SetTemplateLayer](#)

associated entities.

Attaches an annotation reference to one or more associated entities.

Determines whether a block reference is an annotation reference.

Determines whether a block table record is an annotation template.

Determines whether an annotation template has any annotation references.

Determines whether an attribute definition is an annotation text entity.

Refreshes all annotation references in the current drawing that refer to the specified annotation template.

Sets the annotation insert precedence mode.

Sets an expression string to store with an annotation template, annotation reference, or annotation text object.

Sets an expression to store with an annotation template, annotation reference, or annotation text object.

Sets the block color property of an annotation template. See also `GetTemplateColor()`.

Sets the block layer property of an annotation template. See also `GetTemplateLayer()`.

[SetTemplateLinetype](#)

Sets the block linetype property of an annotation template. See also [GetTemplateLinetype\(\)](#).

[SetTemplateLineWeight](#)

Sets the block linewidth property of an annotation template. See also [GetTemplateLineWeight\(\)](#).

[SetTemplateRotation](#)

Sets the block rotation property of an annotation template. See also [GetTemplateRotation\(\)](#).

[SetTemplateScaleFactor](#)

Sets the block scale factor property of an annotation template. See also [GetTemplateScaleFactor\(\)](#).

[TemplateNameBTRPrefix](#)

Returns the prefix that is prepended to all annotation block table records internally.

[UpdateAnnotationReferences](#)

Updates all annotation references in the current drawing that refer to the specified annotation template.

#### ☐ Structures



[AnnotationOverrides](#)

Structure that holds all static-property and expression-string overrides used for creating or updating annotations.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: AnnotationTemplateBlockDefinitionId Function

[AcMapAnnotationManager Namespace](#)

Returns an annotation template's block table record.

```
AcDbObjectId AnnotationTemplateBlockDefinitionId(  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template to examine.

Returns

Returns the object ID of the template's AcDbBlockTableRecord if the template exists; otherwise, returns AcDbObjectId::kNull.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: AnnotationTemplateExists Function

[AcMapAnnotationManager Namespace](#)

Determines whether an annotation template exists.

```
bool AnnotationTemplateExists(  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template to find.
Returns	

Returns true if the template is found in the current drawing; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: AnnotationTemplateReferencedObjIds Function

[AcMapAnnotationManager Namespace](#)

Lists all annotation references of an annotation template in the current drawing.

```
Acad::ErrorStatus AnnotationTemplateReferencedObjIds(  
    AcDbObjectIdArray & objectIdArray,  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
objectIdArray	Output array of object IDs of all the specified template's annotation references.
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: CreateAnnotationTemplate Function

[AcMapAnnotationManager Namespace](#)

Creates an annotation template.

```
AcDbObjectId CreateAnnotationTemplate(  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template to create.

Returns

Returns the object ID of the newly created template if successful; otherwise, returns AcDbObjectId::kNull if an error occurs.

Remarks

If a template with the specified name already exists, it is not overwritten; instead, the null object AcDbObjectId::kNull is returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: DeleteAnnotationTemplate Function

[AcMapAnnotationManager Namespace](#)

Deletes an annotation template.

```
Acad::ErrorStatus DeleteAnnotationTemplate(  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template to delete.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

You can delete a template only if it has no annotation references.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetExpressionString Function

[AcMapAnnotationManager Namespace](#)

Retrieves an expression that is stored with an annotation template, annotation reference, or annotation text object.

```
Acad::ErrorStatus GetExpressionString(  
    ACHAR*& pszExpressionString,  
    const AcDbObject* pObj,  
    AcMapAnnotationManager::eAnnotationExpressionFields field  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszExpressionString	Output expression string stored with the object. The caller must free this object explicitly, typically with <code>acutDelString()</code> .
pObj	Input pointer to the <code>AcDbBlockTableRecord</code> (template), <code>AcDbBlockReference</code> (reference), or <code>AcDbAttributeDefinition</code> (text) to retrieve the expression from.
field	Input <a href="#">eAnnotationExpressionFields</a> value of the expression to retrieve. For annotation templates and annotation references, use only <code>kBlockXxx</code> codes; for annotation text entities, use only <code>kAttDefXxx</code> codes.

Returns

Returns `Acad::eOk` if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateColor Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block color property of an annotation template. See also SetTemplateColor().

```
Acad::ErrorStatus GetTemplateColor(  
    AcCmColor & color,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
color	Output template block color value. This value defaults to the current entity color when a template is created.
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateLayer Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block layer property of an annotation template. See also SetTemplateLayer().

```
Acad::ErrorStatus GetTemplateLayer(  
    ACHAR *& pszLayer,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszLayer	Output template block layer value. This value defaults to the current entity layer when a template is created. The caller must free this object explicitly, typically with acutDelString().
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateLinetype Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block linetype property of an annotation template. See also SetTemplateLinetype().

```
Acad::ErrorStatus GetTemplateLinetype(  
    ACHAR *& pszLinetype,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszLinetype	Output template block linetype value. This value defaults to the current entity linetype when a template is created. The caller must free this object explicitly, typically with acutDelString().
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateLineWeight Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block lineweight property of an annotation template. See also [SetTemplateLineWeight\(\)](#).

```
Acad::ErrorStatus GetTemplateLineWeight(  
    AcDb::Lineweight & lineweight,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
lineweight	Output template block lineweight value. This value defaults to the current entity lineweight when a template is created.
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateName Function

[AcMapAnnotationManager Namespace](#)

Lists all the annotation templates defined in the current drawing.

```
Acad::ErrorStatus GetTemplateName(  
    AcArray<ACHAR *>& pszTemplateNameArray  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateNameArray	Output array of the name of each annotation template defined in the current drawing. The caller must free the memory allocated to this array's template names before destroying the array.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateRotation Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block rotation property of an annotation template. See also SetTemplateRotation().

```
Acad::ErrorStatus GetTemplateRotation(  
    double& dRotation,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
dRotation	Output template block rotation value, expressed in radians. This value defaults to 0.0 when a template is created.
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: GetTemplateScaleFactor Function

[AcMapAnnotationManager Namespace](#)

Retrieves the block scale factor property of an annotation template. See also [SetTemplateScaleFactor\(\)](#).

```
Acad::ErrorStatus GetTemplateScaleFactor(  
    double& dScalefactor,  
    const ACHAR * pszTemplateName  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
dScalefactor	Output template block scale factor value. This value defaults to 1.0 when a template is created.
pszTemplateName	Input name of the template to examine.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReference Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to an associated entity.

```
Acad::ErrorStatus InsertAnnotationReference(  
    AcDbObjectId& newBlockReferenceId,  
    const ACHAR * pszTemplateName,  
    const AcDbObjectId assocEnt,  
    const AcDbBlockReference * pMatchThisBlockReference = NULL  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceId	Input/Output object ID of the newly created annotation reference. If newBlockReferenceId is AcDbObjectId::kNull, then a new annotation reference is created. If not, then the AcDbBlockReference whose Object ID is newBlockReferenceID is reused.
pszTemplateName	Input name of the annotation template to attach.
assocEnt	Input object ID of the entity to which the reference will be attached.
pMatchThisBlockReference	Input optional pointer to an existing annotation reference, whose static-property and expression-string overrides are used to create the new annotation reference.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

This function takes a named annotation template and an associated entity object ID. The new block reference object ID is an output parameter. If the caller

provides an optional pointer to an existing annotation reference, that reference's static-property and expression-string overrides are used to create the new annotation reference.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReference Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to an associated entity.

```
Acad::ErrorStatus InsertAnnotationReference(  
    AcDbObjectId& newBlockReferenceId,  
    const ACHAR * pszTemplateName,  
    const AcDbObjectId assocEnt,  
    const AnnotationOverrides & overrides  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceId	Input/Output object ID of the newly created annotation reference. If newBlockReferenceId is AcDbObjectId::kNull, then a new annotation reference is created. If not, then the AcDbBlockReference whose Object ID is newBlockReferenceID is reused.
pszTemplateName	Input name of the annotation template to attach.
assocEnt	Input object ID of the entity to which the reference will be attached.
overrides	Input AnnotationOverridesstruct containing the static-property and expression-string overrides to be used when creating the new annotation reference. In the AnnotationOverrides structure, pass a NULL for each individual property that you do not want to override; instead, these property values will be computed from the template's corresponding expression or static property.

## Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

## Remarks

This function takes a named annotation template and an associated entity object ID. The new block reference object ID is an output parameter. The static-property and expression-string overrides of the input structure are used to create the new annotation reference (unless a NULL prevents an override).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReferences Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to one or more associated entities.

```
Acad::ErrorStatus InsertAnnotationReferences(  
    AcDbObjectIdArray& newBlockReferenceIds,  
    const ACHAR * pszTemplateName,  
    const AcDbObjectIdArray& assocEntArray,  
    const AcDbBlockReference * pMatchThisBlockReference = NULL  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceIds	Output array of object IDs of the newly created annotation references.
pszTemplateName	Input name of the annotation template to attach.
assocEntArray	Input array of object IDs of the entities to which the references will be attached.
pMatchThisBlockReference	Input optional pointer to an existing annotation reference, whose static-property and expression-string overrides are used to create the new annotation reference.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

This function takes a named annotation template and an array of associated entity object IDs. An array of new block reference object IDs is an output parameter. If the caller provides an optional pointer to an existing annotation reference, that reference's static-property and expression-string overrides are used to create the new annotation references.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: IsAnnotationTemplate Function

[AcMapAnnotationManager Namespace](#)

Determines whether a block table record is an annotation template.

```
bool IsAnnotationTemplate(  
    ACHAR*& pszTemplateName,  
    AcDbBlockTableRecord* pBTR  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Output name of the template if the specified block table record is an annotation template; otherwise, this parameter is not modified. The caller must free this object explicitly, typically with <code>acutDelString()</code> .
pBTR	Input pointer to the block table record to examine.
Returns	

Returns true if the block table record is an annotation template in the current drawing; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: IsAnnotationTemplateReferenced Function

[AcMapAnnotationManager Namespace](#)

Determines whether an annotation template has any annotation references.

```
bool IsAnnotationTemplateReferenced(  
    const ACHAR * pszTemplateName  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the annotation template to examine.
Returns	

Returns true if the template has any references; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::RefreshAnnotationReferences Function

[AcMapAnnotationManager Namespace](#)

Refreshes all annotation references in the current drawing that refer to the specified annotation template.

```
Acad::ErrorStatus RefreshAnnotationReferences(  
    const ACHAR * pszTemplateName,  
    bool bFullAnnotation  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the annotation template to examine.
bFullAnnotation	Input that specifies how the annotation references are refreshed. If true, all expressions of the annotation text entities within the reference(s) are re-evaluated. If false, only the annotation text string is re-evaluated, causing the annotation text values to change but other properties to remain the same.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

Call this function after an entity with an attached annotation reference changes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetExpressionString Function

[AcMapAnnotationManager Namespace](#)

Sets an expression to store with an annotation template, annotation reference, or annotation text object.

```
Acad::ErrorStatus SetExpressionString(  
    const ACHAR* pszExpressionString,  
    AcDbObject* pObj,  
    AcMapAnnotationManager::eAnnotationExpressionFields field  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszExpressionString	Input expression to store with the object.
pObj	Input pointer to the AcDbBlockTableRecord (template), AcDbBlockReference (reference), or AcDbAttributeDefinition (text) to store the expression with.
field	Input <a href="#">eAnnotationExpressionFields</a> value of the expression to store. For annotation templates and annotation references, use only kBlockXxx codes; for annotation text entities, use only kAttDefXxx codes.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateColor Function

[AcMapAnnotationManager Namespace](#)

Sets the block color property of an annotation template. See also GetTemplateColor().

```
Acad::ErrorStatus SetTemplateColor(  
    const ACHAR * pszTemplateName,  
    AcCmColor color  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
color	Input new value for the block color.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateLayer Function

[AcMapAnnotationManager Namespace](#)

Sets the block layer property of an annotation template. See also GetTemplateLayer().

```
Acad::ErrorStatus SetTemplateLayer(  
    const ACHAR * pszTemplateName,  
    const ACHAR * pszLayer  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
pszLayer	Input new value for the block layer.
Returns	

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateLinetype Function

[AcMapAnnotationManager Namespace](#)

Sets the block linetype property of an annotation template. See also GetTemplateLinetype().

```
Acad::ErrorStatus SetTemplateLinetype(  
    const ACHAR * pszTemplateName,  
    const ACHAR * pszLinetype  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
pszLinetype	Input new value for the block linetype.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateLineWeight Function

[AcMapAnnotationManager Namespace](#)

Sets the block linewidth property of an annotation template. See also [GetTemplateLineWeight\(\)](#).

```
Acad::ErrorStatus SetTemplateLineWeight(  
    const ACHAR * pszTemplateName,  
    AcDb::Lineweight linewidth  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
linewidth	Input new value for the block linewidth.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateRotation Function

[AcMapAnnotationManager Namespace](#)

Sets the block rotation property of an annotation template. See also GetTemplateRotation().

```
Acad::ErrorStatus SetTemplateRotation(  
    const ACHAR * pszTemplateName,  
    double dRotation  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
dRotation	Input new value for the block rotation, expressed in radians.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetTemplateScaleFactor Function

[AcMapAnnotationManager Namespace](#)

Sets the block scale factor property of an annotation template. See also [GetTemplateScaleFactor\(\)](#).

```
Acad::ErrorStatus SetTemplateScaleFactor(  
    const ACHAR * pszTemplateName,  
    double dScalefactor  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the template whose static property to set.
dScalefactor	Input new value for the block scale factor.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::UpdateAnnotationReferences Function

[AcMapAnnotationManager Namespace](#)

Updates all annotation references in the current drawing that refer to the specified annotation template.

```
Acad::ErrorStatus UpdateAnnotationReferences(  
    const ACHAR * pszTemplateName,  
    bool bRetainLocalOverrides  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pszTemplateName	Input name of the annotation template to examine.
bRetainLocalOverrides	Input that specifies the manner in which the annotation references are updated. If true, all override expressions contained in each annotation reference are retained during the update. If false, the local overrides are discarded, and the annotation template's expressions and static properties are used.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

Call this function after an annotation template changes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Classification

### ▣ Namespaces

[AcMapObjClass](#) Namespace for classification error codes.

### ▣ Classes

[AcMapClassificationManager](#) Performs general classification operations at the drawing level.

[AcMapObjClassDefinition](#) Manages a feature class definition stored in the feature-definition file.

[AcMapObjClassProperty](#) Manages a property for a feature class definition.

[AcMapObjClassReactor](#) Base class that is notified of classification events.

[AcMapObjClassSystem](#) `AcMapObjClassApi.lib`  
Manages the registration and unregistration of classification reactors.

Links

[Classification](#)

AcMapObjClass Namespace

[Classification](#)

Namespace for classification error codes. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

☐ Enumerations

📄 [EErrCode](#) Classification error codes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClass Namespace](#)

AcMapObjClass:: EErrCode Enumeration

[AcMapObjClass Namespace](#)

Classification error codes.

```
enum EErrCode {
    eOk = 0x1,
    eFailed,
    eFileAlreadyExists,
    eFileNotFound,
    eFileNameInvalid,
    eNoSchemaFileAttached,
    eSchemaFileAttachedAndFound,
    eSchemaFileAttachedButNotFound,
    eOutOfRange,
    eMissingProperty,
    eClassNotFound,
    eBaseClassNotFound,
    eClassAlreadyExists,
    eClassNameInvalid,
    eClassNameTooLong,
    eBaseClassOnly,
    eUnsupportedEntityType,
    eRxCClassNotFound,
    eAlreadyClassified,
    eAlreadyUnclassified,
    eEntityNotClassified,
    ePropertyNotFound,
    ePropertyAlreadyExists,
    eFailedSavingSchema,
    eInvalidType,
    eClassUnsupportedCreateType,
    eProductUnsupportedCreateType,
    eNoUserPrivilegeToAlterSchema,
    eClassNotFromCurrentSchema,
    ePropertyReadOnly
};
```

File

AcMapObjClass.h

Parameters

eOk

Description

The action completed successfully.

eFailed	The action failed.
eFileAlreadyExists	A file with the same name already exists.
eFileNotFound	Could not find the file at the specified location.
eFileNameInvalid	The specified file name is invalid.
eNoSchemaFileAttached	No feature-definition file is attached to the drawing.
eSchemaFileAttachedAndFound	The feature-definition file is attached to the drawing and was found at the specified location.
eSchemaFileAttachedButNotFound	The feature definition-file is attached to the drawing but was not found at the specified location.
eOutOfRange	At least one property value is out of range.
eMissingProperty	At least one property is missing.
eClassNotFound	Could not find the specified feature class in the feature-definition file.
eBaseClassNotFound	Could not find the specified base feature class in the feature-definition file.
eClassAlreadyExists	The specified feature class already exists in the feature-definition file.
eClassNameInvalid	The specified feature class name is invalid.
eClassNameTooLong	The specified feature class name must have 256 or fewer characters.
eBaseClassOnly	The specified feature class cannot be used to classify because it is a strict base class.
eUnsupportedEntityType	The feature class does not support the specified entity type.
eRxClassNotFound	No registered AcRxClass corresponds to the specified entity type.
eAlreadyClassified	The entity is already classified, preventing reclassification.
eAlreadyUnclassified	The entity is already unclassified and cannot be unclassified again.

eEntityNotClassified	The entity is unclassified. No classification-related action can be performed on this entity.
ePropertyNotFound	Could not find the specified property in the feature class definition.
ePropertyAlreadyExists	The specified property already exists in the feature class definition.
eFailedSavingSchema	Could not save the feature-definition file.
eInvalidType	Found an invalid type of property value or an invalid entity type.
eClassUnsupportedCreateType	The specified feature class definition does not support the create method.
eProductUnsupportedCreateType	The product does not support the create method for any feature class definition.
eNoUserPrivilegeToAlterSchema	The current user lacks the privileges to change the feature-definition file.
eClassNotFromCurrentSchema	The <a href="#">AcMapObjClassDefinition</a> instance comes from a feature-definition file other than the current one. The instance is unusable until the original feature-definition file is reattached to the current drawing.
ePropertyReadOnly	The property is a read-only property whose value cannot be changed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classification](#)

Classes

[Classification](#)

☐ Classes

[AcMapClassificationManager](#)

Performs general classification operations at the drawing level. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

[AcMapObjClassDefinition](#)

Manages a feature class definition stored in the feature-definition file. Because all information related to a feature class definition is stored in the feature-definition file, it is possible to manage an AcMapObjClassDefinition instance only if the correct feature-definition file is attached to the current drawing. If the feature-definition file is missing, these functions will fail. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

[AcMapObjClassProperty](#)

Manages a property for a feature class definition. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

[AcMapObjClassReactor](#)

Base class that is notified of classification events. Custom reactors are classes derived from AcMapObjClassReactor. See the AcMapObjClassSystemclass for a description of adding and registering a custom reactor.

[AcMapObjClassSystem](#)

AcMapObjClassApi.lib  
Manages the registration and unregistration of classification reactors. Reactors are classes derived from [AcMapObjClassReactor](#).  
To add a custom reactor:Derive a custom class AcMapObjClassMyReactor from [AcMapObjClassReactor](#):

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapClassificationManager Class

[Classes](#)

Performs general classification operations at the drawing level. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

```
class AcMapClassificationManager;
```

File

AcMapClassificationManager.h

☐ Methods

◆ [~AcMapClassificationManager](#)

Destroys an instance of this class.

◆ [AcMapClassificationManager](#)

Constructs an instance of this class by using the specified AutoCAD database.

[AttachFeatureDefinitionFile](#)

Attaches an existing feature-definition file to the current drawing.

[Audit](#)

Examines an entity for out-of-range or missing classified-property values, fixing these values if desired.

[Audit](#)

Examines multiple entities for out-of-range or missing classified-property values, listing and fixing these values if desired.

[CanCurrentUserAlterSchema](#)

Determines whether the current user has sufficient privileges to change the classification information in the feature-definition file.

[Classify](#)

Classifies an entity with a feature class name.

[Classify](#)

Classifies multiple entities with a feature class name.

Unclassifies an entity entirely.

Unclassifying entities does not trigger

[ClearAllTags](#)

internal transactions. This function differs from Unclassify() because it ignores the feature-definition file that was used to classify the entity.

[ClearAllTags](#)

Unclassifies multiple entities entirely. Unclassifying entities does not trigger internal transactions. This function differs from Unclassify() because it ignores the feature-definition file that was used to classify the entities.

[CreateFeatureClassDefinition](#)

Creates a new feature class definition and adds it to the feature-definition file attached to the current drawing.

[CreateFeatureClassDefinition](#)

Creates a new feature class definition and adds it to the feature-definition file attached to the current drawing.

[CreateFeatureDefinitionFile](#)

Creates a new feature-definition file and attaches it to the current drawing.

[DeleteFeatureClassDefinition](#)

Deletes a feature class definition from the feature-definition file attached to the current drawing.

[DetachCurrentFeatureDefinitionFile](#)

Detaches the current feature-definition file from the current drawing.

[DuplicateFeatureClassDefinition](#)

Creates a copy of an existing feature class definition.

[GetAllTags](#)

Lists all the classification tags of an entity.

[GetClassifiedEntities](#)

Lists all the classified entities in the current drawing.

[GetClassifiedEntities](#)

Lists all the classified entities in the current drawing for a feature class name.

[GetClassifiedProperties](#)

Lists all the classified properties and their values of an entity.

[GetClassifiedProperties](#)

Lists all the classified properties and their values of an entity.

[GetFeatureClassDefinition](#)

Retrieves an existing feature class

<a href="#"><u>GetFeatureClassDefinitionCount</u></a>	definition. Counts the number of feature class definitions in the feature-definition file attached to the current drawing.
<a href="#"><u>GetFeatureClassNames</u></a>	Lists all the feature class definition names in the feature-definition file attached to the current drawing.
<a href="#"><u>GetFeatureDefinitionFileAttached</u></a>	Retrieves the name of the feature-definition file, if any, attached to the current drawing.
<a href="#"><u>GetProperties</u></a>	Lists all the properties and their values of an entity.
<a href="#"><u>GetProperties</u></a>	Lists all the properties and their values of an entity.
<a href="#"><u>GetUnclassifiedEntities</u></a>	Lists all the unclassified entities in the current drawing.
<a href="#"><u>GetUndefinedEntities</u></a>	Lists all the undefined entities in the current drawing.
<a href="#"><u>IsClassified</u></a>	Determines whether an entity is classified.
<a href="#"><u>IsFeatureClassDefinitionPresent</u></a>	Determines whether a class name is defined in the feature-definition file attached to the current drawing.
<a href="#"><u>ReloadCurrentFeatureDefinitionFile</u></a>	Reloads the feature-definition file attached to the current drawing.
<a href="#"><u>RenameFeatureClassDefinition</u></a>	Renames a feature class definition in the feature-definition file attached to the current drawing.
<a href="#"><u>SaveCurrentFeatureDefinitionFile</u></a>	Saves the feature-definition file attached to the current drawing.
<a href="#"><u>SaveCurrentFeatureDefinitionFileAs</u></a>	Saves with a different name the feature-definition file attached to the current drawing.
	Unclassifies an entity. Unclassifying entities does not trigger internal transactions. Unclassify only entities

## Unclassify

classified with the currently attached feature-definition file. If you unclassify an entity that was classified with a different feature-definition file, the corresponding tags will not be changed. To remove all existing tags, regardless of which feature-definition file was used, use `ClearAllTags()`.

Unclassifies multiple entities.

Unclassifying entities does not trigger internal transactions. Unclassify only entities classified with the currently attached feature-definition file. If you unclassify an entity that was classified with a different feature-definition file, the corresponding tags will not be changed. To remove all existing tags, regardless of which feature-definition file was used, use `ClearAllTags()`.

## Unclassify

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: AttachFeatureDefinitionFile Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Attaches an existing feature-definition file to the current drawing.

```
AcMapObjClass::EErrCode AttachFeatureDefinitionFile(  
    const ACHAR* pszSchemaFileName  
);
```

Parameters	Description
pszSchemaFileName	Input name of the feature-definition file to attach. If no path is specified, the current path is used.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eFileNameInvalid if the file name is invalid.  
Returns [AcMapObjClass::EErrCode](#) eFileNotFound if the file does not exist.  
Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

If a feature-definition file is currently attached, it will be automatically replaced.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager::Classify Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Classifies an entity with a feature class name.

```
AcMapObjClass::EErrCode Classify(  
    AcDbObjectId& entId,  
    const ACHAR* pszClassName,  
    bool bIncludeNonConforming,  
    bool bClassifyEvenIfAlreadyClassified  
);
```

Parameters	Description
entId	Input ID of the entity to classify.
pszClassName	Input name of the feature class.
bIncludeNonConforming	Input true to include entities with missing or out-of-range properties, or false to include only conforming properties.
bClassifyEvenIfAlreadyClassified	Input true to replace the current classification if the entity is already classified, or false to prevent reclassification.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eBaseClassOnly if the class is a strict base class that cannot be used for classification. Returns [AcMapObjClass::EErrCode](#) eUnsupportedEntityType if the feature class does not support the specified entity type. Returns [AcMapObjClass::EErrCode](#) eAlreadyClassified if the entity is already classified, preventing reclassification. Returns [AcMapObjClass::EErrCode](#) eOutOfRange if at least one property value is out of range. Returns [AcMapObjClass::EErrCode](#) eMissingProperty if the entity is missing at least one property. Returns [AcMapObjClass::EErrCode](#)

eFailed if the process failed for some other reason.

## Remarks

Classifying entities does not trigger internal transactions.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::Classify Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Classifies multiple entities with a feature class name.

```
AcMapObjClass::EErrCode Classify(  
    AcDbObjectIdArray* paFailedEntIds,  
    AcArray<AcMapObjClass::EErrCode>* paFailedErrCodes,  
    AcDbObjectIdArray& aEntIds,  
    const ACHAR* pszClassName,  
    bool bIncludeNonConforming,  
    bool bClassifyEvenIfAlreadyClassified  
);
```

Parameters	Description
paFailedEntIds	Output IDs of failed entities, or NULL if not needed by the caller. The operation traverses aEntIds from beginning to the end, reporting failures.
paFailedErrCodes	Output error code of each failed entity, or NULL if not needed by the caller. The array index of each error code matches the index of each corresponding failed entity in paFailedEntIds.
aEntIds	Input IDs of the entities to classify.
pszClassName	Input name of the feature class.
bIncludeNonConforming	Input true to include entities with missing or out-of-range properties, or false to include only conforming properties.
bClassifyEvenIfAlreadyClassified	Input true to replace the current classification if the entity is already classified, or false to prevent reclassification.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if all entities are classified successfully. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns

[AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eBaseClassOnly if the class is a strict base class that cannot be used for classification. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for at least one entity.

#### Remarks

Classifying entities does not trigger internal transactions.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::CreateFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Creates a new feature class definition and adds it to the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrCode CreateFeatureClassDefinition(  
    AcMapObjClassDefinition*& pFeatureClassDef,  
    const ACHAR* pszName,  
    const ACHAR* pszDerivedFromClassName,  
    const AcArray<AcRxClass*>& aSupportedEntityTypes,  
    const AcMapStringArray& aBlockNames,  
    bool bBaseClassOnly  
);
```

Parameters	Description
pFeatureClassDef	Output <a href="#">AcMapObjClassDefinition</a> feature class. The caller must free this object.
pszName	Input feature class name.
pszDerivedFromClassName	Input superclass name, or NULL if no base class is needed.
aSupportedEntityTypes	Input array of AcRxClass objects defining the supported entity types. This list typically matches the value that AcRxObject::desc() returns.
aBlockNames	Input array of supported block names.
bBaseClassOnly	Input true to create a strict base class; otherwise, false. You can derive other classes from a strict base class but not classify objects with it.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassAlreadyExists if a class with same name already exists. Returns [AcMapObjClass::EErrCode](#) eBaseClassNotFound if the base class is not in the feature-definition file. Returns

[AcMapObjClass::EErrCode](#) eUnsupportedEntityType if the feature class does not support the specified entity type. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

#### Remarks

A feature class definition can support any block reference, or specific block reference names. For any block reference, the corresponding entity type `AcDbBlockReference::desc()` is input in `aSupportedEntityTypes`. For specific block reference names, the names are input in `aBlockNames`.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::CreateFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Creates a new feature class definition and adds it to the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrCode CreateFeatureClassDefinition(  
    AcMapObjClassDefinition*& pFeatureClassDef,  
    const ACHAR* pszName,  
    const ACHAR* pszDerivedFromClassName,  
    const AcMapStringArray& aSupportedEntityTypes,  
    const AcMapStringArray& aBlockNames,  
    bool bBaseClassOnly  
);
```

Parameters	Description
pFeatureClassDef	Output <a href="#">AcMapObjClassDefinition</a> feature class. The caller must free this object.
pszName	Input feature class name.
pszDerivedFromClassName	Input superclass name, or NULL if no base class is needed.
aSupportedEntityTypes	Input array of strings defining the supported entity types (such as "AcDbCircle" for circle entities). This list typically matches the value that <a href="#">AcRxClass::name()</a> returns.
aBlockNames	Input array of supported block names.
bBaseClassOnly	Input true to create a strict base class; otherwise, false. You can derive other classes from a strict base class but not classify objects with it.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassAlreadyExists if a class with same name already exists. Returns [AcMapObjClass::EErrCode](#) eBaseClassNotFound if the

base class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eUnsupportedEntityType if the feature class does not support the specified entity type. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

#### Remarks

A feature class definition can support any block reference, or specific block reference names. For any block reference, the corresponding entity type `AcDbBlockReference::desc()->name()` is input in `aSupportedEntityTypes`. For specific block reference names, the names are input in `aBlockNames`.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: CreateFeatureDefinitionFile Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Creates a new feature-definition file and attaches it to the current drawing.

```
AcMapObjClass::EErrCode CreateFeatureDefinitionFile(  
    const ACHAR* pszSchemaFileName  
);
```

Parameters	Description
pszSchemaFileName	Input name and full path of the feature-definition file to create.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eFileNameInvalid if the file name is invalid. Returns [AcMapObjClass::EErrCode](#) eFileAlreadyExists if a file with the same name already exists. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: DeleteFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Deletes a feature class definition from the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrCode DeleteFeatureClassDefinition(  
    const ACHAR* pszClassName  
);
```

Parameters	Description
pszClassName	Input name of the feature class to delete.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

Entities classified with the deleted feature class are not updated; instead, they retain their classification but become undefined.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::DuplicateFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Creates a copy of an existing feature class definition.

```
AcMapObjClass::EErrCode DuplicateFeatureClassDefinition(  
    AcMapObjClassDefinition*& pFeatureClassDefCopied,  
    const ACHAR* pszName,  
    const ACHAR* pszCopyName  
);
```

Parameters	Description
pFeatureClassDefCopied	Output new <a href="#">AcMapObjClassDefinition</a> feature class definition. The caller must free this object.
pszName	Input name of the feature class definition to copy.
pszCopyName	Input name of the new feature class definition.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the feature class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eClassAlreadyExists if a class with same name already exists. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager:: GetClassifiedEntities Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the classified entities in the current drawing for a feature class name.

```
bool GetClassifiedEntities(  
    AcDbObjectIdArray& aEntIds,  
    const ACHAR* pszClassName,  
    bool bReturnMachingDerivedClassToo  
) const;
```

Parameters	Description
aEntIds	Output IDs of classified entities.
pszClassName	Input name of the feature class.
bReturnMachingDerivedClassToo	Input true to also retrieve entities classified with classes derived from pszClassName, or false to ignore derived classes. Derived-class functionality is available only if a feature-definition file is attached to the current drawing. (The feature-definition file stores the class hierarchy; only the class name resides on the entity.)

## Returns

Returns true if at least one entity ID is returned; otherwise, returns false if no entity ID is returned or the process failed.

## Remarks

An entity is considered to be classified even if the corresponding feature class definition is not in the attached feature-definition file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetClassifiedProperties Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the classified properties and their values of an entity.

```
AcMapObjClass::EErrCode GetClassifiedProperties(  
    AcArray<AcMapObjClassProperty*>& aProperties,  
    AcArray<VARIANT>* paValues,  
    const AcDbEntity * pEntity,  
    const ACHAR* pszClassName  
) const;
```

Parameters	Description
aProperties	Output array of <a href="#">AcMapObjClassProperty</a> properties. The caller must free this object.
paValues	Output array of property values, or NULL if not needed by the caller. The array index of each property value matches the index of each corresponding property in aProperties. The caller must free this object.
pEntity	Input entity to examine.
pszClassName	Input name of the feature class.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns  
[AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid.  
Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is  
too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no  
feature-definition file is attached to the current drawing. Returns  
[AcMapObjClass::EErrCode](#) eEntityNotClassified if the entity is unclassified.  
Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the entity is classified  
but the class is not in the feature-definition file. Returns  
[AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this  
message disappear you need to register this software. If you have problems  
registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetClassifiedProperties Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the classified properties and their values of an entity.

```
AcMapObjClass::EErrCode GetClassifiedProperties(  
    AcArray<AcMapObjClassProperty*>& aProperties,  
    AcArray<VARIANT>* paValues,  
    const AcDbObjectId& entId,  
    const ACHAR* pszClassName  
) const;
```

Parameters	Description
aProperties	Output array of <a href="#">AcMapObjClassProperty</a> properties. The caller must free this object.
paValues	Output array of property values, or NULL if not needed by the caller. The array index of each property value matches the index of each corresponding property in aProperties. The caller must free this object.
entId	Input ID of the entity to examine.
pszClassName	Input name of the feature class.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns  
[AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid.  
Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is  
too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no  
feature-definition file is attached to the current drawing. Returns  
[AcMapObjClass::EErrCode](#) eEntityNotClassified if the entity is unclassified.  
Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the entity is classified  
but the class is not in the feature-definition file. Returns  
[AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this  
message disappear you need to register this software. If you have problems  
registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Retrieves an existing feature class definition.

```
AcMapObjClass::EErrCode GetFeatureClassDefinition(  
    AcMapObjClassDefinition*& pFeatureClassDef,  
    const ACHAR* pszName  
);
```

Parameters	Description
pFeatureClassDef	Output <a href="#">AcMapObjClassDefinition</a> feature class definition. The caller must free this object.
pszName	Input name of the feature class definition to retrieve.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the feature class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetFeatureDefinitionFileAttached Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Retrieves the name of the feature-definition file, if any, attached to the current drawing.

```
AcMapObjClass::EErrCode GetFeatureDefinitionFileAttached(  
    ACHAR*& pszSchemaFileName  
) const;
```

Parameters	Description
pszSchemaFileName	Output the full pathname of the attached feature-definition file, or NULL if no file is attached. The caller must free this object, typically with acutDelString().

## Returns

Returns [AcMapObjClass::EErrCode](#) eSchemaFileAttachedAndFound if the feature-definition file is attached to the current drawing and was found. Returns [AcMapObjClass::EErrCode](#) eSchemaFileAttachedButNotFound if the feature-definition file is attached to the current drawing but was not found. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

It is possible for a feature-definition file to be attached to the current drawing but physically missing from the indicated location (path); use the returned error code to determine the file status.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager:: IsClassified Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Determines whether an entity is classified.

```
bool IsClassified(  
    AcMapStringArray& aClassNames,  
    const AcDbObjectId& entId,  
    const ACHAR* pszSchemaFileName  
) const;
```

Parameters	Description
aClassNames	Output array of class names, or empty if the entity is unclassified. This array typically contains one item because the classification feature permits only a single classification for a given feature-definition file.
entId	Input ID of the entity to examine.
pszSchemaFileName	Input name of the feature-definition file. This value, which determines the class definition's location, can be the name of the current or a now-detached feature-definition file.

## Returns

Returns true if the entity is classified for the specified feature-definition file; otherwise, returns false if the entity is unclassified.

## Remarks

An entity is considered to be classified even if the corresponding feature class definition is not in the attached feature-definition file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager:: IsFeatureClassDefinitionPresent Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Determines whether a class name is defined in the feature-definition file attached to the current drawing.

```
bool IsFeatureClassDefinitionPresent(  
    const ACHAR* pszClassName  
) const;
```

Parameters	Description
pszClassName	Input name of the feature class.

### Returns

Returns true if the class name is defined; otherwise, returns false if it is undefined or no feature-definition file is attached to the current drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::RenameFeatureClassDefinition Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Renames a feature class definition in the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrCode RenameFeatureClassDefinition(  
    const ACHAR* pszClassName,  
    const ACHAR* pszNewClassName  
);
```

Parameters	Description
pszClassName	Input current name of the feature class.
pszNewClassName	Input new name of the feature class.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eClassAlreadyExists if a class with the same name already exists. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name is invalid. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name is too long. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

The new name must be a non-empty string with 256 or fewer characters. The valid characters are: a-z A-Z 0-9 \$ \_ - (including accented characters). Entities classified with the renamed feature class are not updated; instead, they become undefined.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: SaveCurrentFeatureDefinitionFileAs Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Saves with a different name the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrCode SaveCurrentFeatureDefinitionFileAs(  
    const ACHAR* pszSchemaFileName  
) const;
```

Parameters	Description
pszSchemaFileName	Input name of the new feature-definition file. If no path is specified, the current path is used.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eFileAlreadyExists if a file with the same name already exists. Returns [AcMapObjClass::EErrCode](#) eFailedSavingSchema if the feature-definition file could not be saved. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

This process simply duplicates the existing file; it does not replace it or attach it to current drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapObjClassDefinition Class

[Classes](#)

Manages a feature class definition stored in the feature-definition file. Because all information related to a feature class definition is stored in the feature-definition file, it is possible to manage an AcMapObjClassDefinition instance only if the correct feature-definition file is attached to the current drawing. If the feature-definition file is missing, these functions will fail. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

```
class AcMapObjClassDefinition;  
File
```

AcMapObjClassDefinition.h

☐ Methods



[~AcMapObjClassDefinition](#)

Destroys an instance of this class.

[AddProperty](#)

Adds a new property to this feature class definition.

[DeleteProperty](#)

Deletes a property from this feature class definition.

[GetCreateMethod](#)

Retrieves the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

[GetCreateMethodName](#)

Retrieves the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

[GetDescription](#)

Retrieves the description of this feature class definition.

[GetDirectBaseClassName](#)

Retrieves the direct base class name of this feature class definition.

[GetFeatureDefinitionFile](#)

Retrieves the full pathname of the feature-definition file that this feature class definition belongs to.

<a href="#"><u>GetIconName</u></a>	Retrieves the icon name of this feature class definition.
<a href="#"><u>GetName</u></a>	Retrieves the name of this feature class definition.
<a href="#"><u>GetProperties</u></a>	Lists all the classified properties of this feature class definition.
<a href="#"><u>GetProperty</u></a>	Retrieves a property of this feature class definition.
<a href="#"><u>GetSupportedEntityTypes</u></a>	Lists the AutoCAD entity types that this feature class definition supports.
<a href="#"><u>GetSupportedEntityTypes</u></a>	Lists the AutoCAD entity types that this feature class definition supports.
<a href="#"><u>IsBaseClassOf</u></a>	Determines whether this feature class definition is a base class of the specified class, either directly or deeper in the class hierarchy.
<a href="#"><u>IsBaseClassOnly</u></a>	Determines whether this feature class definition is a strict base class.
<a href="#"><u>IsDerivedClassOf</u></a>	Determines whether this feature class definition is derived from the specified class, either directly or from higher in the class hierarchy.
<a href="#"><u>IsDirectBaseClassOf</u></a>	Determines whether this feature class definition is the direct base class of the specified class.
<a href="#"><u>IsPropertyClassified</u></a>	Determines whether a property of this feature class definition is classified.
<a href="#"><u>IsVisibleInWorkspace</u></a>	Determines whether this feature class definition is visible in the AutoCAD Map project workspace.
<a href="#"><u>LinkedDataMovedToFdo</u></a>	This function allows a client program to find out whether a link to the external data is going to be preserved to the FDO or the whole record is conveyed to the FDO.
<a href="#"><u>PropertyLinkTemplate</u></a>	Determines whether a property of this feature class definition is a link template property and returns the link template name.
<a href="#"><u>SetCreateMethod</u></a>	Sets the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.
	Sets the AutoCAD entity type that this feature

[SetCreateMethod](#)

class definition uses when a digitize process runs for this class.

[SetDescription](#)

Sets the description of this feature class definition.

[SetIconName](#)

Sets the icon name of this feature class definition.

[SetLinkedDataMovedToFdo](#)

This function allows a client program to define the behavior of the linked data in process of querying and saving from/to FDO. A link to the external data only will be preserved to the FDO if the boolean parameter is set to true. Otherwise, records will be moved to the FDO. conveyed to the FDO.

[SetName](#)

Sets the name of this feature class definition.

[SetVisibleInWorkspace](#)

Sets this feature class definition to visible or invisible in the AutoCAD Map project workspace.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::DeleteProperty Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Deletes a property from this feature class definition.

```
AcMapObjClass::EErrCode DeleteProperty(  
    const AcMapStringArray& aStrParentToSubCategoryNames,  
    const ACHAR* pszPropertyName  
) const;
```

Parameters

Description

aStrParentToSubCategoryNames Input category of the property. This array must contain the hierarchy of all categories for the property, from the top base category to the last sub-category that is the property's direct category.

pszPropertyName

Input name of the property to delete.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) ePropertyNotFound if the property does not exist. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetCreateMethod Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

```
AcMapObjClass::EErrCode GetCreateMethod(  
    AcRxClass*& prxEntityType,  
    ACHAR*& pszBlockName  
) const;
```

Parameters	Description
prxEntityType	Output entity type defining the create method, or NULL if undefined.
pszBlockName	Output block name, or NULL if undefined or no block is used. The caller must free this object, typically with <code>acutDelString()</code> .

## Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if successful. Returns [AcMapObjClass::EErrCode](#) `eClassNotFromCurrentSchema` if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for some other reason.

## Remarks

The entity type typically matches the value that `AcRxObject::desc()` returns. If the feature class definition uses a specific block reference name as a create method, `AcDbBlockReference::desc()` is output in `prxEntityType` and the name is output in `pszBlockName`.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: GetCreateMethodName Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

```
AcMapObjClass::EErrCode GetCreateMethodName(  
    ACHAR*& pszEntityType,  
    ACHAR*& pszBlockName  
) const;
```

Parameters	Description
pszEntityType	Output entity type defining the create method, or NULL if undefined. The caller must free this object, typically with acutDelString().
pszBlockName	Output block name, or NULL if undefined or no block is used. The caller must free this object, typically with acutDelString().

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

The entity type typically matches the value that AcRxClass::name() returns. If the feature class definition uses a specific block reference name as a create method, AcDbBlockReference::desc()->name() is output in pszEntityType and the name is output in pszBlockName.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetDescription Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the description of this feature class definition.

```
AcMapObjClass::EErrCode GetDescription(  
    ACHAR*& pszDescription
```

```
) const;
```

Parameters

Description

pszDescription

Output description, or an empty string if no description is set for the class. The caller must free this object, typically with `acutDelString()`.

Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if successful. Returns

[AcMapObjClass::EErrCode](#) `eClassNotFromCurrentSchema` if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: GetDirectBaseClassName Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the direct base class name of this feature class definition.

```
AcMapObjClass::EErrCode GetDirectBaseClassName(  
    const ACHAR*& pszBaseClassName  
) const;
```

Parameters	Description
pszBaseClassName	Output the name of direct base class.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetIconName Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the icon name of this feature class definition.

```
AcMapObjClass::EErrCode GetIconName(  
    ACHAR*& pszIconName  
) const;
```

Parameters	Description
pszIconName	Output full pathname of the icon's bitmap file, or an empty string if no icon name is set (meaning that the class is using the standard icon). The caller must free this object, typically with acutDelString().

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::GetProperty Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves a property of this feature class definition.

```
AcMapObjClass::EErrCode GetProperty(  
    AcMapObjClassProperty*& pProperty,  
    const AcMapStringArray& aStrParentToSubCategoryNames,  
    const ACHAR* pszPropertyName  
) const;
```

Parameters	Description
pProperty	Output <a href="#">AcMapObjClassProperty</a> property object. The caller must free this object.
aStrParentToSubCategoryNames	Input category. This array must contain the hierarchy of all categories for the property, from the top base category to the last sub-category that is the property's direct category.
pszPropertyName	Input name of the property to retrieve.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) ePropertyNotFound if the property does not exist. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsBaseClassOf Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether this feature class definition is a base class of the specified class, either directly or deeper in the class hierarchy.

[AcMapObjClass::EErrCode](#) IsBaseClassOf(  
    **bool**\* pbIsBaseClass,  
    **const** ACHAR\* pszName  
) **const**;

Parameters

Description

pbIsBaseClass

Output true if this feature class definition is a direct or indirect base class of the specified class; otherwise, false.

pszName

Input name of the feature class definition to examine.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns

[AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsDerivedClassOf Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether this feature class definition is derived from the specified class, either directly or from higher in the class hierarchy.

```
AcMapObjClass::EErrorCode IsDerivedClassOf(  
    bool* pbIsDerivedClass,  
    const ACHAR* pszName  
) const;
```

Parameters	Description
pbIsDerivedClass	Output true if this feature class definition is a descendant of the specified class; otherwise, false.
pszName	Input name of the feature class definition to examine.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsDirectBaseClassOf Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether this feature class definition is the direct base class of the specified class.

```
AcMapObjClass::EErrCode IsDirectBaseClassOf(  
    bool* pbIsDirectBaseClass,  
    const ACHAR* pszName  
) const;
```

Parameters	Description
pbIsDirectBaseClass	Output true if this feature class definition is a direct base class of the specified class; otherwise, false.
pszName	Input name of the feature class definition to examine.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsPropertyClassified Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether a property of this feature class definition is classified.

```
AcMapObjClass::EErrorCode IsPropertyClassified(  
    bool* pbIsPropertyClassified,  
    const AcMapStringArray& aStrParentToSubCategoryNames,  
    const ACHAR* pszPropertyName  
) const;
```

Parameters	Description
pbIsPropertyClassified	Output true if the property is classified; otherwise, false.
aStrParentToSubCategoryNames	Input category of the property. This array must contain the hierarchy of all categories for the property, from the top base category to the last sub-category that is the property's direct category.
pszPropertyName	Input name of the property to examine.
Returns	

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::PropertyLinkTemplate Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether a property of this feature class definition is a link template property and returns the link template name.

```
AcMapObjClass::EErrCode PropertyLinkTemplate(  
    ACHAR*& pcLinkTemplate,  
    const AcMapStringArray& aStrParentToSubCategoryNames,  
    const ACHAR* pszPropertyName  
) const;
```

Parameters	Description
pcLinkTemplate	Output link template name if the property is LinkTemplate related , otherwise NULL.
aStrParentToSubCategoryNames	Input category of the property. This array must contain the hierarchy of all categories for the property, from the top base category to the last sub-category that is the property's direct category.
pszPropertyName	Input name of the property to examine.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: SetCreateMethod Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

```
AcMapObjClass::EErrCode SetCreateMethod(  
    const ACHAR* pszEntityType,  
    const ACHAR* pszBlockName  
);
```

Parameters	Description
pszEntityType	Input entity type.
pszBlockName	Input block name.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eInvalidType if the entity type is invalid. This error code also is returned if pszBlockName is empty or NULL, even if pszEntityType is AcDbBlockReference::desc()->name(); a name is required to define a valid block create method. Returns [AcMapObjClass::EErrCode](#) eClassUnsupportedCreateType if the feature class definition does not support the create method. Returns [AcMapObjClass::EErrCode](#) eProductUnsupportedCreateType if the product does not support the create method. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

The entity type typically matches the value that AcRxClass::name() returns. Invalid types generate an error. The entity types that the digitize process supports are only a subset of the valid entity types, so a type that the feature class definition supports can still be invalid. If a feature class definition uses a specific block reference name as a create method, set pszBlockName to the name and set pszEntityType to NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::SetCreateMethod Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets the AutoCAD entity type that this feature class definition uses when a digitize process runs for this class.

```
AcMapObjClass::EErrCode SetCreateMethod(  
    const AcRxClass* prxEntityType,  
    const ACHAR* pszBlockName  
);
```

Parameters	Description
prxEntityType	Input entity type.
pszBlockName	Input block name.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eInvalidType if the entity type is invalid. This error code also is returned if pszBlockName is empty or NULL, even if pszEntityType is AcDbBlockReference::desc(); a name is required to define a valid block create method. Returns [AcMapObjClass::EErrCode](#) eClassUnsupportedCreateType if the feature class definition does not support the create method. Returns [AcMapObjClass::EErrCode](#) eProductUnsupportedCreateType if the product does not support the create method. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

The entity type typically matches the value that AcRxObject::desc() returns. Invalid types generate an error. The entity types that the digitize process supports are only a subset of the valid entity types, so a type that the feature class definition supports can still be invalid. If a feature class definition uses a specific block reference name as a create method, set pszBlockName to the name and set prxEntityType to NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: SetDescription Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets the description of this feature class definition.

```
AcMapObjClass::EErrCode SetDescription(  
    const ACHAR* pszDescription  
);
```

Parameters	Description
pszDescription	Input description.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: SetIconName Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets the icon name of this feature class definition.

```
AcMapObjClass::EErrorCode SetIconName(  
    const ACHAR* pszIconName  
);
```

Parameters

Description

pszIconName

Input full pathname of the icon's bitmap file, or NULL or an empty string to specify the standard icon.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::SetName Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets the name of this feature class definition.

```
AcMapObjClass::EErrCode SetName(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input class name.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFound if the class is not in the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eClassAlreadyExists if a class with the same name already exists. Returns [AcMapObjClass::EErrCode](#) eClassNameInvalid if the class name contains invalid characters. Returns [AcMapObjClass::EErrCode](#) eClassNameTooLong if the class name contains more than 255 characters. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

If you change the name of an existing feature class definition, entities so classified are not updated; instead, they become undefined.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapObjClassProperty Class

[Classes](#)

Manages a property for a feature class definition. For more information, search for *feature classification* and *feature definitions* in AutoCAD Map Help.

```
class AcMapObjClassProperty;
```

File

AcMapObjClassProperty.h

☐ Methods



<a href="#">~AcMapObjClassProperty</a>	Destroys an instance of this class.
<a href="#">FromString</a>	Generates a value from a string representation.
<a href="#">GetCategory</a>	Retrieves the direct category of this property.
<a href="#">GetDefaultValue</a>	Retrieves the default value of this property.
<a href="#">GetName</a>	Retrieves the name of this property.
<a href="#">GetRange</a>	Retrieves the range of valid values for this property.
<a href="#">GetType</a>	Retrieves the type of this property.
<a href="#">GetValue</a>	Retrieves the value of this property from the specified entity.
<a href="#">IsInRange</a>	Determines whether a value falls in this property's range of valid values.
<a href="#">IsReadOnly</a>	Determines whether this property is read-only or read-write.
<a href="#">IsVisible</a>	Determines whether this property is visible or invisible.
<a href="#">SetDefaultValue</a>	Sets the default value of this property.
<a href="#">SetRange</a>	Sets or unsets the range of valid values for this property.
<a href="#">SetReadOnly</a>	Sets this property to read-only or read-write.
	Sets the value of this property for the specified entity. The entity and property can be classified or unclassified. If this property is classified, this function checks the property's range of valid values and, if the value is

### [SetValue](#)

invalid, sets the value to its default value if `bFixOutOfRangeIfClassified` is true. Properties retrieved with [AcMapClassificationManager::GetClassifiedProperties](#) (two forms) are classified properties. Properties retrieved with [AcMapClassificationManager::GetProperties](#) (two forms) are unclassified properties, and `SetValue` sets property values regardless of the value of `bFixOutOfRangeIfClassified`.

### [SetVisible](#)

Sets this property to visible or invisible.

### [ToString](#)

Returns the string representation of a value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty::FromString Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Generates a value from a string representation.

```
void FromString(  
    VARIANT* pvarValue,  
    const ACHAR* pszValue  
) const;
```

Parameters	Description
pvarValue	Output value. The caller must free this object, typically with VariantClear().
pszValue	Input string.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: SetRange Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Sets or unsets the range of valid values for this property.

```
AcMapObjClass::EErrCode SetRange(  
    const ACHAR* pszRange  
);
```

Parameters	Description
pszRange	Input range of valid values, or NULL or "--" to unset the range.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the property's class is not in the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eOutOfRange if the current default value falls outside of the specified range. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapObjClassReactor Class

[Classes](#)

Base class that is notified of classification events. Custom reactors are classes derived from AcMapObjClassReactor. See the AcMapObjClassSystemclass for a description of adding and registering a custom reactor.

```
class AcMapObjClassReactor;
```

File

AcMapObjClassReactor.h

☐ Methods

<a href="#">~AcMapObjClassReactor</a>	Destroys an instance of this class.
<a href="#">AcMapObjClassReactor</a>	Constructs an instance of this class.
<a href="#">FeatureClassDefinitionCreated</a>	Invoked when a feature class definition is created.
<a href="#">FeatureClassDefinitionDeleted</a>	Invoked when a feature class definition is deleted.
<a href="#">FeatureClassDefinitionModified</a>	Invoked when a feature class definition is modified.
<a href="#">FeatureClassDefinitionRenamed</a>	Invoked when a feature class definition is renamed.
<a href="#">FeatureDefinitionFileAttached</a>	Invoked when a feature-definition file is attached to or detached from the current drawing.
<a href="#">FeatureDefinitionFileModified</a>	Invoked when a feature-definition file is modified and saved.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

AcMapObjClassReactor::FeatureClassDefinitionCreated Method

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature class definition is created.

```
virtual void FeatureClassDefinitionCreated(  
    const ACHAR * pszFileName,  
    const ACHAR * pszClassName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file.
pszClassName	Input name of the feature class definition.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

AcMapObjClassReactor::FeatureClassDefinitionDeleted Method

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature class definition is deleted.

```
virtual void FeatureClassDefinitionDeleted(  
    const ACHAR * pszFileName,  
    const ACHAR * pszClassName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file.
pszClassName	Input name of the feature class definition.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

AcMapObjClassReactor::FeatureClassDefinitionModified Method

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature class definition is modified.

```
virtual void FeatureClassDefinitionModified(  
    const ACHAR * pszFileName,  
    const ACHAR * pszClassName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file.
pszClassName	Input name of the feature class definition.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

AcMapObjClassReactor::FeatureClassDefinitionRenamed Method

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature class definition is renamed.

```
virtual void FeatureClassDefinitionRenamed(  
    const ACHAR * pszFileName,  
    const ACHAR * pszOldClassName,  
    const ACHAR * pszNewClassName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file.
pszOldClassName	Input old name of the feature class definition.
pszNewClassName	Input new name of the feature class definition.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

[AcMapObjClassReactor::FeatureDefinitionFileAttached Method](#)

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature-definition file is attached to or detached from the current drawing.

```
virtual void FeatureDefinitionFileAttached(  
    const ACHAR * pszFileName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file that was attached, or an empty string if the file was detached.

## Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)

AcMapObjClassReactor::FeatureDefinitionFileModified Method

[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Invoked when a feature-definition file is modified and saved.

```
virtual void FeatureDefinitionFileModified(  
    const ACHAR * pszFileName  
);
```

Parameters	Description
pszFileName	Input name and full path of the feature-definition file.

Returns

Returns nothing.

## Remarks

Typically, notification occurs at the moment changes are committed to a file by saving it. Note that a file can be saved with no changes, which still triggers notification.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapObjClassSystem Class

[Classes](#)

AcMapObjClassApi.lib

Manages the registration and unregistration of classification reactors. Reactors are classes derived from [AcMapObjClassReactor](#).

To add a custom reactor: Derive a custom class AcMapObjClassMyReactor from [AcMapObjClassReactor](#):

```
class AcMapObjClassMyReactor : public AcMapObjClassReactor
```

Implement events by overriding the virtual functions that you need. Create an instance of the custom reactor:

```
AcMapObjClassMyReactor* pMyReactor = new AcMapObjClassMyReactor;
```

Register it so that it becomes active:

```
AcMapObjClassSystem().AddObjClassReactor(pMyReactor);
```

Write some classification code. Remove the reactor from the list and delete it:

```
AcMapObjClassSystem().RemoveObjClassReactor(pMyReactor);  
delete pMyReactor;
```

Note that only one list of reactors exists independently of the drawing, so if you change classification information in a feature-definition file, all the drawings are notified of the event.

```
class AcMapObjClassSystem;  
File
```

AcMapObjClassSystem.h

☐ Methods



[~AcMapObjClassSystem](#)

Destroys an instance of this class.



[AcMapObjClassSystem](#)

Constructs an instance of this class.

[AddObjClassReactor](#)

Adds a reactor to the list of the classification reactors.

[RemoveObjClassReactor](#)

Removes a reactor from the list of the classification reactors.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Create Centroids

### ▣ Namespaces

#### [AcMapUtilities](#)

This namespace provides functions that create centroids.

Links

[Create Centroids](#)

AcMapUtilities Namespace

[Create Centroids](#)

This namespace provides functions that create centroids. Newly created centroids contain the same object data and link template information, if any, of the source entities for which the centroids were created. For more information, search for *centroids* in AutoCAD Map Help.

☐ Functions

[CreateCentroids](#)

Creates centroid blocks inside a set of closed polylines or mpolygons.

[CreateCentroids](#)

Creates centroid blocks inside a set of closed polylines or mpolygons.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapUtilities Namespace](#)

AcMapUtilities::CreateCentroids Function

[AcMapUtilities Namespace](#)

Creates centroid blocks inside a set of closed polylines or mpolygons.

```
AcMap::EErrCode CreateCentroids(  
    AcDbObjectIdArray& aCentroidIds,  
    AcDbObjectIdArray& aClosedPolyIds,  
    const ACHAR* pszLayerName,  
    const ACHAR* pszBlockName  
);  
File
```

AcMapUtilities.h

Parameters	Description
aCentroidIds	Output array that contains the identifiers of all the created centroids.
aClosedPolyIds	Input array that contains the identifiers of the closed polylines or mpolygons.
pszLayerName	Input name of the layer to create the centroids on.
pszBlockName	Input name of block to use for the created centroids.

Returns

Returns AcMap::kOk if successful. Returns AcMap::kErrBadInput if the named block does not exist. Returns AcMap::kErr if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapUtilities Namespace](#)

AcMapUtilities:: CreateCentroids Function

[AcMapUtilities Namespace](#)

Creates centroid blocks inside a set of closed polylines or mpolygons.

```
AcMap::EErrorCode CreateCentroids(  
    ads_name& ssCentroids,  
    ads_name& ssClosedPolys,  
    const ACHAR* pszLayerName,  
    const ACHAR* pszBlockName  
);  
File
```

AcMapUtilities.h

Parameters	Description
ssCentroids	Output name of the ADS (AutoCAD Development System) selection set that contains all the created centroids.
ssClosedPolys	Input name of the ADS selection set that contains the closed polylines or mpolygons.
pszLayerName	Input name of the layer to create the centroids on.
pszBlockName	Input name of block to use for the created centroids.
Returns	

Returns AcMap::kOk if successful. Returns AcMap::kErrBadInput if the named block does not exist. Returns AcMap::kErr if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Custom Object Protocol Extensions

### ☐ Classes

#### [AcMapGeometryPE](#)

A set of functions to analyze custom entities as collections of simple geometry elements (points, polylines, polygons).

#### [AcMapQueryPE](#)

A set of functions to analyze custom entities for querying, altering their properties when they are queried, and saving them back to their source drawings.

Links

[Custom Object Protocol Extensions](#)

Classes

[Custom Object Protocol Extensions](#)

☐ Classes

[AcMapGeometryPE](#)

A set of functions to analyze custom entities as collections of simple geometry elements (points, polylines, polygons).

[AcMapQueryPE](#)

A set of functions to analyze custom entities for querying, altering their properties when they are queried, and saving them back to their source drawings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapGeometryPE Class

[Classes](#)

A set of functions to analyze custom entities as collections of simple geometry elements (points, polylines, polygons).

```
class AcMapGeometryPE : public AcRxObject;
```

File

AcMapGeometryPE.h

Remarks

Valid polygons are closed figures composed of a single contour only. A polygon containing another polygon is read as two polygons, not as a polygon with a hole. Polygons are processed as solid objects during a query. If a geometry element is not a point, polyline, or polygon, its type is unsupported. Unsupported types are ignored.

☐ Enumerations

 [EGeometryType](#) Types for simple geometry elements.

☐ Methods

[beginRead](#) Executes before reading an entity begins.

[endRead](#) Executes after reading an entity ends.

[getGeometryType](#) Gets the type of a simple element.

[getNumGeometries](#) Counts simple elements in an entity.

[getNumVertices](#) Counts vertices of a polyline or polygon.

[getPoint](#) Gets a point element.

[getVertex](#) Gets a vertex of a polyline or polygon.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapQueryPE Class

[Classes](#)

A set of functions to analyze custom entities for querying, altering their properties when they are queried, and saving them back to their source drawings.

```
class AcMapQueryPE : public AcRxObject;
```

File

AcMapQueryPE.h

☐ Enumerations

📄 [ELocationType](#) Location-query process types.

📄 [EPropType](#) Property types for AutoCAD Map entities.

☐ Methods

[canQuery](#) Can an entity be queried?.

[canSaveBack](#) Can the entity be saved back?.

[getPointsOnEntity](#) Get any physical point on the entity to check whether it is inside location boundary or outside.

[getPropertyValue](#) Gets a property value of an entity.

[hasProperty](#) Does the entity have a given property?.

[isClosed](#) One more property alteration functions telling whether we need to apply hatch alteration to the entity or not.

[isPropertyReadOnly](#) Can an entity be changed by property alteration?.

[locationQueryType](#) Specifies how the entity will be processed in a location query.

[onClosedSet](#) Sends a notification to the entity at the time of closed set calculation (query and save back).

[setPropertyValue](#) Performs a property alteration on an entity.

[swapIdWith](#) Swap ID with the prototype entity during saving back.

[transformBy](#) Transforms the entity (projection transformation during query and save back, and rubber-sheeting).

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE:: canQuery Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Can an entity be queried?.

```
virtual bool canQuery(  
    AcDbEntity* pEnt = NULL,  
    AcDbDatabase* toDb = NULL  
);
```

Parameters	Description
pEnt	The entity.
toDb	Database to query to.

Returns

Returns true if the entity can be queried.

Remarks

entity can be NULL, meaning you are being asked about all entities of this type.  
database will be also NULL in this case.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Data Sources

### ▣ Classes

#### [AcMapDataSources](#)

This class provides methods that handle data sources.

Links

[Data Sources](#)

Classes

[Data Sources](#)

☐ Classes

This class provides methods that handle data sources.

[AcMapDataSources](#)

For more information, search for *data sources* in AutoCAD Map Help.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDataSources Class

[Classes](#)

This class provides methods that handle data sources. For more information, search for *data sources* in AutoCAD Map Help.

**class** AcMapDataSources;

File

MapDataSources.h

☐ Methods

◆ [~AcMapDataSources](#)

Destroys an instance of this class.

◆ [AcMapDataSources](#)

Constructs an instance of this class.

[AttachDataSource](#)

Attaches a data source to the current AutoCAD Map work session.

[ConnectDataSource](#)

Connects a data source to the current AutoCAD Map work session.

[DetachAllDataSources](#)

Detaches all data sources from the current AutoCAD Map work session.

[DetachDataSource](#)

Detaches a data source from the current AutoCAD Map work session.

[DisconnectAllDataSources](#)

Disconnects all data sources from the current AutoCAD Map work session.

[DisconnectDataSource](#)

Disconnects a data source from the current AutoCAD Map work session.

[GetAttachedDataSources](#)

Retrieves the names of the attached data sources in the current AutoCAD Map work session.

[GetAttachedDataSourcesCount](#)

Counts the number of attached data sources in the current AutoCAD Map work session.

[GetConnectedDataSources](#)

Retrieves the names of the connected data sources in the current AutoCAD Map work session.

[GetConnectedDataSourcesCount](#) Counts the number of connected data sources in the current AutoCAD Map work session.

[GetDisconnectedDataSources](#) Retrieves the names of the disconnected data sources in the current AutoCAD Map work session.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

[AcMapDataSources:: AttachDataSource Method](#)

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Attaches a data source to the current AutoCAD Map work session.

```
bool AttachDataSource(  
    ACHAR * pszDataSourceName,  
    const ACHAR * pszDataSourcePath  
);
```

Parameters	Description
pszDataSourceName	Output name of the data source that was attached.
pszDataSourcePath	Input absolute pathname of the data source to attach. Relative pathnames are not allowed.

Returns

Returns true if the data source is attached, or false if it is not attached.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

AcMapDataSources:: ConnectDataSource Method

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Connects a data source to the current AutoCAD Map work session.

```
bool ConnectDataSource(  
    const ACHAR* pszDataSourceName  
);
```

Parameters	Description
pszDataSourceName	Input name of the data source to connect.

Returns

Returns true if the data source is connected, or false if it not connected.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

AcMapDataSources::DetachDataSource Method

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Detaches a data source from the current AutoCAD Map work session.

```
void DetachDataSource(  
    const ACHAR* pszDataSourceName  
);
```

Parameters	Description
pszDataSourceName	Input name of the data source to detach.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)  
AcMapDataSources:: DisconnectDataSource Method  
[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Disconnects a data source from the current AutoCAD Map work session.

```
void DisconnectDataSource(  
    const ACHAR* pszDataSourceName  
);
```

Parameters	Description
pszDataSourceName	Input name of the data source to disconnect.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Display Manager

### ▣ Classes

#### [AcMapDMAllDrawOrderItemsIterator](#)

An iterator over the draw order of elements (AcMapDMElement). The iterator will step on every element, including those which are currently in a proxied state.

#### [AcMapDMAllItemsIterator](#)

An iterator over a collect of display-management items. This iterator will step on every item, including those which may be in a proxy state.

#### [AcMapDMAllStyleReferencesIterator](#)

An iterator over a collect of style references (AcMapDMStyleReferer) within one style set. The iterator may return a StyleId() that points to an AcDbProxyObject.

#### [AcMapDMAttachedDwgsQueryDataSourceDescriptor](#)

Describes a display-management ADE query and provides functions to define queries and manage drawing lists.

#### [AcMapDMAttachedDwgsQueryElement](#)

Represents a display-management ADE query element in an attached drawing.

#### [AcMapDMCurrentDwgQueryElement](#)

Represents a query element in the current drawing.

#### [AcMapDMDataSourceDescriptor](#)

Represents a query definition's data-source descriptor.

#### [AcMapDMDefaultStyle](#)

Represents the default

[AcMapDMElement](#)

[AcMapDMEntityStyle](#)

[AcMapDMFeatureDataSourceDescriptor](#)

[AcMapDMFeatureElement](#)

[AcMapDMGroup](#)

[AcMapDMItem](#)

[AcMapDMLayerDataSourceDescriptor](#)

[AcMapDMLayerElement](#)

[AcMapDMLegend](#)

[AcMapDMMap](#)

[AcMapDMMapIterator](#)

[AcMapDMMapManager](#)

stylization.

An abstraction of an AutoCAD Map element that manipulates associated styles. A concrete element provides its own implementation for object selection.

Encapsulates properties that can be applied to entities

Represents a feature-class data source.

Represents a feature-class element.

Represents a group of display-management items (

Base class of all display-management items.

Represents a layer data source.

Represents a layer element

Represents a legend object

Represents the Display Manager map, the root object from which all other Display Manager objects can be obtained by using iterator.

An iterator over a collection of display-management items (

Manages display-management map objects (AcMapDMMap).

Base class used to notify

[AcMapDMMapReactor](#)

application of display-management events with a display-management map.

[AcMapDMProjectReactor](#)

Base class used to notify an application of display-management events with an AutoCAD Map project.

[AcMapDMQueryDataSourceDescriptor](#)

Describes a base data source for querying.

[AcMapDMRasterDataSourceDescriptor](#)

Defines a raster data source.

[AcMapDMRasterElement](#)

Represents a raster element that selects raster images.

[AcMapDMRasterStyle](#)

Encapsulates raster-image related properties.

[AcMapDMSEAnnotationStyle](#)

Provides an annotation alteration style for stylization.

[AcMapDMSEHatchStyle](#)

Provides a hatch alteration style for stylization.

[AcMapDMSETextStyle](#)

Provides a text alteration style for stylization.

[AcMapDMStyle](#)

Base class for display-management styles.

[AcMapDMStyleCategory](#)

Represents a category of styles.

[AcMapDMStyleLibrary](#)

Represents a library of categories of styles.

[AcMapDMStyleReference](#)

Represents a reference to existing style.

[AcMapDMStylizationEntityStyle](#)

Represents a style of a stylization entity.

[AcMapDMThematicStyle](#)

Represents a thematic stylization.

Defines a display-

[AcMapDMThematicTable](#)

management thematic table and its items.

[AcMapDMThematicTableItemAnnotation](#)

Represents an annotation cell of a thematic definition table.

[AcMapDMThematicTableItemBlock](#)

Represents a block cell of a thematic definition table.

[AcMapDMThematicTableItemColor](#)

Represents a color cell of a thematic definition table.

[AcMapDMThematicTableItemDataValue](#)

Represents a text/value-pair cell of a thematic definition table.

[AcMapDMThematicTableItemHatch](#)

Represents a hatch cell of a thematic definition table.

[AcMapDMThematicTableItemLegendText](#)

Represents a legend text cell of a thematic definition table.

[AcMapDMThematicTableItemLinestyle](#)

Represents a linestyle cell of a thematic definition table.

[AcMapDMThematicTableItemLinetype](#)

Represents a linetype cell of a thematic definition table.

[AcMapDMThematicTableItemLineweight](#)

Class representing a lineweight cell of a thematic definition table.

[AcMapDMThematicTableItemPlotstyle](#)

Represents a plotstyle cell of a thematic definition table.

[AcMapDMThematicTableItemText](#)

Represents a text cell of a thematic definition table.

[AcMapDMThematicTableRow](#)

Utility class that defines a thematic table row and allows easy access to its column items.

[AcMapDMTopoElement](#)

Defines a query for all topological elements in the current drawing.

[AcMapDMTopoElementDataSourceDescriptor](#)

Defines a topology data-source.

[AcMapDMTopoQueryDataSourceDescriptor](#)

Defines a topology data-source.

[AcMapDMTopoQueryElement](#)

Defines a query element that acquires topology objects in the source drawing.

Links

[Display Manager](#)

AcMapDMScaleFactor Namespace

[Display Manager](#)

Provides scale-related information and reactor methods.

☐ Functions

[DMAddScaleFactorReactor](#)

Adds a scale-factor reactor to monitor scale-factor changes.

[DMDisplayIsMetric](#)

Determines the display measurement units.

[DMDwgUnitsAreMetric](#)

Determines the dwg measurement units.

[DMGetCurrentScaleFactor](#)

Retrieves the cached current scale factor.

[DMGetUnCachedCurrentScaleFactor](#)

Immediately computes the current scale factor.

[DMRefreshScale](#)

Forces the scale factor to update.

[DMRemoveScaleFactorReactor](#)

Removes a scale-factor reactor.

☐ Classes

[DMScaleFactorReactor](#) Reactor class for scale-factor changes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMGetUnCachedCurrentScaleFactor Function

[AcMapDMScaleFactor Namespace](#)

Immediately computes the current scale factor.

```
double DMGetUnCachedCurrentScaleFactor(  
    AcDbDatabase * pDb,  
    double override_dpi = 0.0,  
    long override_windnumber = -1  
);  
File
```

DmScaleFactor.h

Parameters	Description
pDb	Input AcDbDatabase to compute scale factor.
override_dpi	Input value to override internal dpi calculation. Default value of 0.0 indicates to the function to get the screen dpi from the operating system.
override_windnumber	Input value to override selected viewport number for a layout. Default value of -1 indicates to the function to use the current paperspace layout viewport to compute the scale factor

Returns

Returns the current scale factor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapDMDisplayManagement Namespace

[Display Manager](#)

Namespace for display-management utility functions.

☐ Enumerations

 [EDMStatus](#)

Enumerates the types of display-management active status.

☐ Functions

[AddProjectReactor](#)

Adds a display-management reactor to monitor project-level activities such as creation and deletion.

[CommitEdits](#)

Used to commit uncommitted edits to objects the Element queried in from external data sources.

[CommitEdits](#)

Used to commit uncommitted edits to objects the Elements within the specified Map queried in from external data sources.

[DiscardEdits](#)

Used to discard uncommitted edits to objects the Element queried in from external data sources.

[DiscardEdits](#)

Used to discard uncommitted edits to objects the Elements within the specified Map queried in from external data sources.

[DMMapManagerId](#)

Retrieves the ID of the display-management map manager object for a database.

exist

[DMMapManagerId](#)

Retrieves the ID of the display-management map manager object

[DMStyleLibraryId](#)

for a project.

Retrieves the ID of display-management style library object for a database.

[DMStyleLibraryId](#)

Retrieves the ID of display-management style library object for a project.

[GetCurrentMapBaseElementInvisibleEntities](#)

Retrieves the visibility status of the map base element and the IDs of entities made invisible when it was unchecked.

[GetDMMapManager](#)

Retrieves the display-management map manager object for the specified database.

[GetDMMapManager](#)

Retrieves the display-management map manager object for a specified AutoCAD Map project.

[GetDMStyleLibrary](#)

Retrieves the display-management style library object for a database.

[GetDMStyleLibrary](#)

Retrieves the display-management style library object for a project.

[GetEntityIdForFeatureOnLayer](#)

This function will search the current Map in the provided database to see if there is a FeatureEntity for a given GWS FeatureId currently in existence.

[GetGWS](#)

This function opens the GWS that the map refers to.

[GetVectorElementIdForFeatureLayer](#)

This function will search the current Map in the provided database to see if there is a VectorElement for the given Layer name.

[HasDMData](#)

Determines whether a project has display-management data.

[HasUncommittedEdits](#)

Used to determine if an Element has uncommitted edits to objects it queried in from external data sources.

[HasUncommittedEdits](#)

Used to determine if a Map has uncommitted edits to objects queried in from external data sources by any of the Map's Elements.

[OnBeginWritingToDataStore](#)

Notifies the display manager that an application is about to write a drawing database to an external data store.

[OnEndWritingToDataStore](#)

Notifies the display manager that an application is finished writing a drawing database to an external data store.

[RemoveProjectReactor](#)

Removes a display-management reactor from a project.

[Status](#)

Retrieves the display-management current active status.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetEntityIdForFeatureOnLayer Function

[AcMapDMDisplayManagement Namespace](#)

This function will search the current Map in the provided database to see if there is a FeatureEntity for a given GWS FeatureId currently in existence.

```
Acad::ErrorStatus GetEntityIdForFeatureOnLayer(  
    AcDbObjectId& entId,  
    AcDbDatabase* pDb,  
    const TCHAR* layerSourceName,  
    int featureId  
);  
File
```

DmDisplayManagement.h

Parameters	Description
entId	Output the FeatureEntity id. May be kNull.
pDb	The database to search.
layerSourceName	The name of the GWS LayerSource in which the featureId resides.
featureId	The featureId of the GWS feature instance. This should be a GWS 'cache id'.

Returns

Returns Acad::eOk if successful, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetGWS Function

[AcMapDMDisplayManagement Namespace](#)

This function opens the GWS that the map refers to.

```
Acad::ErrorStatus GetGWS(  
    IGWS** pGWS,  
    AcDbDatabase* pDb  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pGWS	Output opened IGWS.
pDatabase	Input database.

Returns

Returns Acad::eOk if successful, Acad::eKeyNotFound if the dictionary entry with the file name does not exist otherwise, returns a different error code.

Remarks

If the GWS does not exist yet it will create it and assign it to the Map. Once it has been created the name of the XML file is stored in the dictionary; this function retrieves the name from the dictionary to nopen the GWS.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetVectorElementIdForFeatureLayer Function

[AcMapDMDisplayManagement Namespace](#)

This function will search the current Map in the provided database to see if there is a VectorElement for the given Layer name.

```
Acad::ErrorStatus GetVectorElementIdForFeatureLayer(  
    AcDbObjectId& elementId,  
    AcDbDatabase* pDb,  
    const TCHAR* layerSourceName  
);
```

File

DmDisplayManagement.h

Parameters	Description
pDb	The database to search.
layerSourceName	The name of the GWS LayerSource in which the featureId resides.
entId	Output the VectorElement id. May be kNull.

Returns

Returns Acad::eOk if successful, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapDMThematicUpdateFlags Enumeration

[Display Manager](#)

This is record AcMapDMThematicUpdateFlags.

```
enum AcMapDMThematicUpdateFlags {  
    kUpdateWithoutText = 0x00,  
    kUpdateTextOnly = 0x04  
};  
File
```

DmThematicStyle.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapThematicTableColumnKey Enumeration

[Display Manager](#)

Enumerates defined column indexes for a thematic definition table.

```
enum AcMapThematicTableColumnKey {  
    kColkey_DataValue = 0,  
    kColkey_LegendText = 1,  
    kColkey_Color = 2,  
    kColkey_Linestyle = 3,  
    kColkey_Block = 4,  
    kColkey_Text = 5,  
    kColkey_Hatch = 6,  
    kColkey_Linetype = 7,  
    kColkey_Lineweight = 8,  
    kColkey_Plotstyle = 9,  
    kColkey_Annotation = 10  
};  
File
```

DmThematicTable.h

Parameters	Description
kColkey_DataValue	Data value.
kColkey_LegendText	Legend text.
kColkey_Color	Color.
kColkey_Linestyle	Linestyle.
kColkey_Block	Block.
kColkey_Text	Text.
kColkey_Hatch	Hatch.
kColkey_Linetype	Linetype.
kColkey_Lineweight	Lineweight.
kColkey_Plotstyle	Plotstyle.
kColkey_Annotation	Annotation.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapThematicTableItemInsertionPoint Enumeration

[Display Manager](#)

Enumerates the insertion point for thematic table items.

```
enum AcMapThematicTableItemInsertionPoint {  
    kCenterIns,  
    kCentroidIns,  
    kLabelptIns,  
    kX1Ins,  
    kX2Ins,  
    kX3Ins,  
    kX4Ins  
};
```

};

File

DmThematicTable.h

Parameters	Description
kCenterIns	Center.
kCentroidIns	Centroid.
kLabelptIns	Label.
kX1Ins	X1.
kX2Ins	X2.
kX3Ins	X3.
kX4Ins	X4.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapThematicTableItemJustificationType Enumeration

[Display Manager](#)

Enumerates justification types for thematic table items.

```
enum AcMapThematicTableItemJustificationType {  
    kLeftJust,  
    kCenterJust,  
    kMiddleJust,  
    kRightJust  
};  
File
```

DmThematicTable.h

Parameters	Description
kLeftJust	Left justification.
kCenterJust	Center justification.
kMiddleJust	Middle justification.
kRightJust	Right justification.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

AcMapDMThematicBuildRangesErrorCode Enumeration

[Display Manager](#)

Error codes that are returned when a build-ranges operation fails in thematic mapping. See also [AcMapDMThematicStyle::BuildRangeTables\(\)](#).

```
enum AcMapDMThematicBuildRangesErrorCode {
    keUnknownError = 0,
    keOk = 1,
    keCouldNotOpenThemeTable,
    keMapSessionIsNull,
    keGetProjectFailed,
    keGetRangeLibFailed,
    keAddRangeTabFailed,
    keTableGetHatchIsNull,
    keTableGetDataIsNull,
    keTableGetColorIsNull,
    keTableGetLinestyleIsNull,
    keTableGetBlockIsNull,
    keTableGetTextIsNull,
    keTableGetLineweightIsNull,
    keTableGetLinetypeIsNull,
    keTableGetPlotstyleIsNull,
    keTableGetAnnotationIsNull
};
```

File

DmThematicStyle.h

Parameters	Description
keUnknownError	Unknown error.
keOk	Completed successfully.
keCouldNotOpenThemeTable	Cannot open the theme table.
keMapSessionIsNull	Null AutoCAD Map session.
keGetProjectFailed	Cannot get AutoCAD Map project.
keGetRangeLibFailed	Cannot get range library.
keAddRangeTabFailed	Cannot add range table.
keTableGetHatchIsNull	Null hatch.
keTableGetDataIsNull	Null data.

keTableGetColorIsNull	Null color.
keTableGetLinestyleIsNull	Null linestyle.
keTableGetBlockIsNull	Null block.
keTableGetTextIsNull	Null text.
keTableGetLineweightIsNull	Null lineweight.
keTableGetLinetypeIsNull	Null linetype.
keTableGetPlotstyleIsNull	Null plotstyle.
keTableGetAnnotationIsNull	Null annotation.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Display Manager](#)

Classes

[Display Manager](#)

☐ Classes

[AcMapDMAllDrawOrderItemsIterator](#)

An iterator over the draw (AcMapDMElement). The element, including those state.

[AcMapDMAllItemsIterator](#)

An iterator over a collect. This iterator will step on may be in a proxy state. (

[AcMapDMAllStyleReferencesIterator](#)

An iterator over a collect (AcMapDMStyleReferer) iterator may return a Styl AcDbProxyObject.

[AcMapDMAttachedDwgsQueryDataSourceDescriptor](#)

Describes a display-man functions that define que

[AcMapDMAttachedDwgsQueryElement](#)

Represents a display-ma attached drawing.

[AcMapDMCurrentDwgQueryElement](#)

Represents a query elem

[AcMapDMDataSourceDescriptor](#)

Represents a query defin

[AcMapDMDefaultStyle](#)

Represents the default sty

[AcMapDMElement](#)

An abstraction of an Aut manipulates associated st

[AcMapDMEntityStyle](#)

Encapsulates properties t

[AcMapDMFeatureDataSourceDescriptor](#)

Represents a feature-clas

[AcMapDMFeatureElement](#)

Represents a feature-clas

[AcMapDMGroup](#)

Represents a group of dis (AcMapDMItem).

[AcMapDMItem](#)

Base class of all display-

[AcMapDMLayerDataSourceDescriptor](#)

Represents a layer data s

[AcMapDMLayerElement](#)

Represents a layer eleme

Represents a legend obje characteristics: A legend active (checked) Themes

[AcMapDMLegend](#)

[AcMapDMMap](#)

[AcMapDMMapIterator](#)

[AcMapDMMapManager](#)

[AcMapDMMapReactor](#)

[AcMapDMProjectReactor](#)

[AcMapDMQueryDataSourceDescriptor](#)

[AcMapDMRasterDataSourceDescriptor](#)

display-manager tree. The matches the order of items. Each "space" (modelspace) has zero or one legend, Map and Scale when a legend object does not stand for a map it is associated with.

Represents the Display Manager which all other Display Managers use using an iterator.

An iterator over a collection of objects (AcMapDMMapManager).

Manages display-manager objects (AcMapDMMap).

Base class used to notify management events with Derive custom reactors from this class.

To add a reactor: Derive from AcMapDMMapReactor and implement the functions of this reactor base class.

[AcMapDMMap::AddReactor](#) registers a reactor. To unregister a reactor, use [AcMapDMMap::RemoveReactor](#).

Base class used to notify management events with Derive custom reactors from this class.

To add a reactor: Derive from AcMapDMProjectReactor and implement the functions of this reactor base class.

[AcMapDMDisplayManager::Register](#) registers an instance of the reactor. Use [AcMapDMDisplayManager::Unregister](#) to unregister.

[AcMapDMDisplayManager::Register](#)

Describes a base data source descriptor.

Defines a raster data source descriptor.

[AcMapDMRasterElement](#)  
[AcMapDMRasterStyle](#)

[AcMapDMSEAnnotationStyle](#)

[AcMapDMSEHatchStyle](#)

[AcMapDMSETextStyle](#)

[AcMapDMStyle](#)

[AcMapDMStyleCategory](#)

Represents a raster element.  
Encapsulates raster-image data.  
Provides an annotation and hatch style.  
Several properties can be set as a value or as an ADE expression. Setting a value clears any value previously set. Likewise, setting an ADE expression clears any value previously set. `GetXxxIsExpression()` is provided to test which state is currently in effect for each property. To represent a value, set the expression to NULL for the wrong `GetXxx` function. [more](#)

Provides a hatch alteration style. Several properties can be represented as an ADE expression (in state for each representation). Setting a value previously set clears any value previously set. Likewise, setting an ADE expression clears any value previously set. `GetXxxIsExpression()` is provided to test which state is currently in effect for each property. To represent a value, set the expression to NULL for the wrong `GetXxx` function for the property.

Provides a text alteration style. Several properties can be represented as an ADE expression (in state for each representation). Setting a value previously set clears any value previously set. Likewise, setting an ADE expression clears any value previously set. `GetXxxIsExpression()` is provided to test which state is currently in effect for each property. To represent a value, set the expression to NULL for the wrong `GetXxx` function for the property.

Base class for display-map style categories.  
Represents a category of display-map style.

<a href="#"><u>AcMapDMStyleLibrary</u></a>	Represents a library of ca
<a href="#"><u>AcMapDMStyleReference</u></a>	Represents a reference to
<a href="#"><u>AcMapDMStylizationEntityStyle</u></a>	Represents a style of a st
<a href="#"><u>AcMapDMThematicStyle</u></a>	Represents a thematic sty
<a href="#"><u>AcMapDMThematicTable</u></a>	Defines a display-manag
<a href="#"><u>AcMapDMThematicTableItemAnnotation</u></a>	Represents an annotation
<a href="#"><u>AcMapDMThematicTableItemBlock</u></a>	Represents a block cell o
<a href="#"><u>AcMapDMThematicTableItemColor</u></a>	Represents a color cell of
<a href="#"><u>AcMapDMThematicTableItemDataValue</u></a>	Represents a text/value-p table.
<a href="#"><u>AcMapDMThematicTableItemHatch</u></a>	Represents a hatch cell o
<a href="#"><u>AcMapDMThematicTableItemLegendText</u></a>	Represents a legend text
<a href="#"><u>AcMapDMThematicTableItemLinestyle</u></a>	Represents a linestyle cel
<a href="#"><u>AcMapDMThematicTableItemLinetype</u></a>	Represents a linetype cel
<a href="#"><u>AcMapDMThematicTableItemLineweight</u></a>	Class representing linewe table.
<a href="#"><u>AcMapDMThematicTableItemPlotstyle</u></a>	Represents a plotstyle ce
<a href="#"><u>AcMapDMThematicTableItemText</u></a>	Represents a text cell of :
<a href="#"><u>AcMapDMThematicTableRow</u></a>	Utility class that defines access to its column item
<a href="#"><u>AcMapDMTopoElement</u></a>	Defines a query for all to drawing.
<a href="#"><u>AcMapDMTopoElementDataSourceDescriptor</u></a>	Defines a topology data-:
<a href="#"><u>AcMapDMTopoQueryDataSourceDescriptor</u></a>	Defines a topology data-:
<a href="#"><u>AcMapDMTopoQueryElement</u></a>	Defines a query element the source drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMAAllDrawOrderItemsIterator Class

[Classes](#)

An iterator over the draw order of elements (AcMapDMElement). This iterator will step on every element, including those which are currently in a proxied state.

```
class AcMapDMAAllDrawOrderItemsIterator;
```

File

DmDisplayItem.h

☐ Methods



[~AcMapDMAAllDrawOrderItemsIterator](#) Destroys an instance of this class.

[Done](#) Determines whether the iterator has reached the end of the collection.

[Next](#) Advances to the next element in the iteration.

[ObjectId](#) Retrieves the ID of the current object in the iteration.

[Rewind](#) Moves to the first element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAAllDrawOrderItemsIterator Class](#),

[AcMapDMAAllDrawOrderItemsIterator Class](#)

AcMapDMAAllDrawOrderItemsIterator:: Rewind Method

[AcMapDMAAllDrawOrderItemsIterator Class](#) |

[AcMapDMAAllDrawOrderItemsIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMAAllItemsIterator Class

[Classes](#)

An iterator over a collection of display-management items This iterator will step on every item, including those which may be in a proxy state. (AcMapDMIItem).

```
class AcMapDMAAllItemsIterator;
```

File

DmDisplayItem.h

☐ Methods



[~AcMapDMAAllItemsIterator](#) Destroys an instance of this class.

[Done](#) Determines whether the iterator has reached the end of the collection.

[Next](#) Advances to the next element in the iteration.

[ObjectId](#) Retrieves the ID of the current object in the iteration.

[Rewind](#) Moves to the first element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllItemsIterator Class](#), [AcMapDMAllItemsIterator Class](#)

[AcMapDMAllItemsIterator::Rewind Method](#)

[AcMapDMAllItemsIterator Class](#) | [AcMapDMAllItemsIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMAllStyleReferencesIterator Class

[Classes](#)

An iterator over a collection of style references ([AcMapDMStyleReference](#)) within one style set. The iterator may return a `StyleId()` that points to an `AcDbProxyObject`.

```
class AcMapDMAllStyleReferencesIterator;
```

File

DmDisplayElement.h

☐ Methods



[~AcMapDMAllStyleReferencesIterator](#) Destroys an instance of this class.

[Done](#) Determines whether the iterator has reached the end of the collection.

[Next](#) Advances to the next element in the iteration.

[ObjectId](#) Retrieves the ID of the current object in the iteration.

[Rewind](#) Moves to the first element in the iteration.

[StyleId](#) Retrieves the ID of the style that the current object points to.

[ThresholdScale](#) Retrieves the scale threshold of the current object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor Class

[Classes](#)

Describes a display-management ADE query and provides functions that define queries and manage drawing lists.

```
class AcMapDMAttachedDwgsQueryDataSourceDescriptor : public AcMapDMQ  
File
```

DmAdeQueryElement.h

Remarks

This class is an abstraction of an "envelope" to communicate selection definition between an element and the corresponding user-interface component.

☐ Methods



[~AcMapDMAttachedDwgsQueryDataSourceDescriptor](#)

Destroys an instance of this class.



[AcMapDMAttachedDwgsQueryDataSourceDescriptor](#)

Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[GetDrawingList](#)

Lists the drawings used as the scope of the query.

[GetQuery](#)

Retrieves the query definition.

[SetAcquisitionStatement](#)

Sets the query string.

[SetDrawingList](#)

Sets the list of drawings to use as the scope of the query.

[SetQuery](#)

Sets the query definition.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor::

GetAcquisitionStatement Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor::

SetAcquisitionStatement Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMAttachedDwgsQueryElement Class

[Classes](#)

Represents a display-management ADE query element in an attached drawing.

**class** AcMapDMAttachedDwgsQueryElement : **public** [AcMapDMCurrentDwgQuer](#)  
File

DmAdeQueryElement.h

Remarks

Use this class to specify query criteria and acquire objects that satisfy them.

☐ Methods



[~AcMapDMAttachedDwgsQueryElement](#) Destroys an instance of this class.



[AcMapDMAttachedDwgsQueryElement](#) Constructs an instance of this class.

[AcquireEntities](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[DismissEntities](#)

Erases entities that are part of this query element.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is

called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfInFields](#)

Lets this object read its data. See also dxInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfOutFields](#)

Lets this object write its data. See also dxOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

### [EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

### [EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

### [GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

### [GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

### [SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

### [SetVisible](#)

Makes the entities that are part of this element visible or invisible.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: EvaluateExpressionValues Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: EvaluateExpressionValues Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: GetAcquisitionCriteria Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters

Description

pszString

Output pointer to the data-source descriptor, as a string representation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMCurrentDwgQueryElement Class

[Classes](#)

Represents a query element in the current drawing.

```
class AcMapDMCurrentDwgQueryElement : public AcMapDMElement;  
File
```

DmProjectQueryElement.h

☐ Methods



[~AcMapDMCurrentDwgQueryElement](#) Destroys an instance of this class.



[AcMapDMCurrentDwgQueryElement](#) Constructs an instance of this class.

[AcquireEntities](#)

Runs the query against the current drawing.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that

[dxfoutfields](#)

you will need to call it directly.

Lets this object write its data. See also dxfoutfields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[OnMapProjectInitialized](#)

Invoked when an AutoCAD Map project is initialized.

[SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: EvaluateExpressionValues Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: EvaluateExpressionValues Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: GetAcquisitionCriteria Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters

Description

pszString

Output pointer to the data-source descriptor, as a string representation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMDataSourceDescriptor Class

[Classes](#)

Represents a query definition's data-source descriptor.

```
class AcMapDMDataSourceDescriptor : public AcRxObject;  
File
```

DmDisplayElement.h

☐ Methods

[GetAcquisitionStatement](#) Retrieves the string representation of the query.

[SetAcquisitionStatement](#) Sets the query string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDataSourceDescriptor Class](#), [AcMapDMDataSourceDescriptor Class](#)  
AcMapDMDataSourceDescriptor:: GetAcquisitionStatement Method  
[AcMapDMDataSourceDescriptor Class](#) | [AcMapDMDataSourceDescriptor Class](#)

Retrieves the string representation of the query.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDataSourceDescriptor Class](#), [AcMapDMDataSourceDescriptor Class](#)  
AcMapDMDataSourceDescriptor:: SetAcquisitionStatement Method  
[AcMapDMDataSourceDescriptor Class](#) | [AcMapDMDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMDefaultStyle Class

[Classes](#)

Represents the default stylization.

```
class AcMapDMDefaultStyle : public AcMapDMStyle;
```

File

DmDefaultStyle.h

☐ Enumerations

 [DefaultStyleType](#) Enumerates the types of default styles.

☐ Methods

 [~AcMapDMDefaultStyle](#) Destroys an instance of this class.

 [AcMapDMDefaultStyle](#) Constructs an instance of this class.

[Apply](#) Applies the style to an entity.

[ClearFade](#) Clears the fade. See also IsFadeCleared().

[clone](#) Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgInFields](#) Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX

[dwgOutFields](#)

Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[GetFade](#)

Retrieves the fade property. See also `SetFade()`.

[GetStyleType](#)

Retrieves the style type. This function is intended for use in future versions of AutoCAD Map, when styles other than `DefaultStyleType::kEntityFade` may be available. See also `SetStyleType()`.

[IsFadeCleared](#)

Determines whether the fade is cleared. See also `ClearFade()`.

[Preview](#)

Applies style traits to yield a preview of the specified entity.

[SetFade](#)

Sets the fade property. In this release of AutoCAD Map, this function simply turns the target entity to ACI color 9 (light gray). Future releases may permit other fade colors. See also `GetFade()`.

[SetStyleType](#)

Sets the style type. This function is intended for use in future versions of AutoCAD Map, when styles other than `DefaultStyleType::kEntityFade` may be available. See also `GetStyleType()`.

[UnApply](#)

Builds stylization information for later regeneration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)  
AcMapDMDefaultStyle:: Preview Method  
[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of created alteration entities.
targetEntId	Input ID of preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatching, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMElement Class

[Classes](#)

An abstraction of an AutoCAD Map element that manipulates associated styles.

```
class AcMapDMElement : public AcMapDMItem;
```

File

DmDisplayElement.h

Remarks

A concrete element provides its own implementation for object selection.

☐ Enumerations



[EPreviewEntityType](#)

Enumerates the preview entity types.

☐ Methods

◆ [~AcMapDMElement](#)

Destroys an instance of this class.

◆ [AcMapDMElement](#)

Constructs an instance of this class.

[AcquireEntities](#)

Runs the query and retrieves and stores the IDs of objects that meet the selection criteria.

[AddStyle](#)

Adds a new style to this element.

[AddStyle](#)

Adds a new style to this element.

[audit](#)

This function is called by AutoCAD when the AUDIT command is executed.

[ClonesObjectsFromExternalSource](#)

Clones objects that are from the external source at AcquireEntities() time.

[DismissEntities](#)

Clears the acquired entities that are stored in this element.

[DismissStylization](#)

Clears the current stylization.

Gets an alternative, constant name

[DisplayName](#)

for this type of element in certain situations where the instance name may be unavailable.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxInFields](#)

Lets this object read its data. See also `dxInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxOutFields](#)

Lets this object write its data. See also `dxOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EnableStyle](#)

Enables the style reference's style.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

This function may be called by derived classes during their

[ExcludeEntitiesFromStylization](#)

AcquireEntities() implementation, if the Element wishes to explicitly exclude certain entities from being stylized under all circumstances.

[FilterOutStylizationEntities](#)

Filters out those entities from the selection that, during stylization, acquired stylization entities of another element.

[GetAcquiredEntities](#)

Retrieves the current selection (defined by calling AcquireEntities()).

[GetAcquisitionCriteria](#)

Retrieves a string representation of the acquisition criteria.

[GetAcquisitionCriteria](#)

Retrieves the acquisition criteria for this element.

[GetAllStyleReferencesIterator](#)

Retrieves an iterator over a collection of style references (for the corresponding style set) applied to this element.

[GetCustomPreviewEntityName](#)

Retrieves the custom preview entity name. See also [SetCustomPreviewEntityName\(\)](#).

[GetPreviewBlockId](#)

Retrieves the AcDbObjectId of the block used for preview, if preview is of type kPreviewBlock. See also [SetPreviewBlockId\(\)](#).

[GetPreviewEntityType](#)

Retrieves the preview entity type of this element. See also [SetPreviewEntityType\(\)](#).

[GetTitle](#)

Retrieves the name of this element. See also [SetTitle\(\)](#).

[HasStyleReference](#)

Determines whether this element has a specified style reference.

[IsStyleApplied](#)

Determines whether this element has a style associated with it at a specified scale.

<a href="#"><u>IsStylized</u></a>	Determines whether this element is stylized.
<a href="#"><u>IsTopElement</u></a>	Determines if this element is the top element in the map.
<a href="#"><u>IsUniqueStyleSet</u></a>	Determines whether the style set on this element is unique. See also <code>MakeStyleSetUnique()</code> .
<a href="#"><u>IsVisible</u></a>	Determines whether this element is visible at the specified scale. See also <code>SetVisible()</code> .
<a href="#"><u>MakeStyleSetUnique</u></a>	Makes unique the style set associated with this element. See also <code>IsUniqueStyleSet()</code> .
<a href="#"><u>MoveStyle</u></a>	Moves a style within an element.
<a href="#"><u>OnMapProjectInitialized</u></a>	Invoked when an AutoCAD Map project is initialized.
<a href="#"><u>OnObjectAppended</u></a>	Invoked when an object is appended to the database.
<a href="#"><u>OrderEntitiesByDrawOrder</u></a>	Order an array of objects based on their current draw order.
<a href="#"><u>PlaceEntitiesInFrontOfDrawOrder</u></a>	Place this array of objects at the top of draw order.
<a href="#"><u>QueueObjectsForRegenerate</u></a>	Puts all queried objects that are part of this element on the regeneration queue. This function may be called by derived classes which used the <code>excludeEntityFromStylization()</code> method.
<a href="#"><u>RemoveEntityFromAcquisition</u></a>	Removes the indicated <code>entityId</code> from the Element's acquisition set.
<a href="#"><u>RemoveStyle</u></a>	Removes a style from this element.
<a href="#"><u>RemoveStyle</u></a>	Removes a style from this element.
<a href="#"><u>RemoveStyle</u></a>	Removes a style from this element.
<a href="#"><u>Render</u></a>	Stylize and Render layer contents to the specified renderer.

[SetAcquiredEntities](#)

Sets the current selection (defined by calling `AcquireEntities()`).

[SetAcquisitionCriteria](#)

Sets the acquisition criteria for this element.

[SetCustomPreviewEntityName](#)

Sets the name for `kPreviewBlock` and `kPreviewCustom` preview entity types. See also [GetCustomPreviewEntityName\(\)](#).

[SetPreviewBlockId](#)

Sets the `AcDbObjectId` of the block for `kPreviewBlock` preview entity type. See also `GetPreviewBlockId()`.

[SetPreviewEntityType](#)

Sets the preview entity type. See also [GetPreviewEntityType\(\)](#).

[SetTitle](#)

Sets the name of this element. See also `GetTitle()`.

[SetVisible](#)

Sets the element visibility at the specified scale. See also `IsVisible()`.

[subErase](#)

Invoked from within `erase()` before the erase actually occurs. See also `subErase()` in the AutoCAD ObjectARX Developer's Guide.

[UpdateStylization](#)

Stylizes the current or updated selection at the current scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: AddStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Adds a new style to this element.

```
virtual Acad::ErrorStatus AddStyle(  
    AcDbObjectId& styleRefId,  
    AcMapDMStyle* pStyle,  
    AcMapDMAAllStyleReferencesIterator& Position  
);
```

Parameters	Description
styleRefId	Output ID of the added style reference.
pStyle	Input pointer to a AcMapDMStyleobject.
Position	Input position at which to add the style. See also <a href="#">AcMapDMAAllStyleReferencesIterator</a>

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The style object and the owning AcMapDMMap object must be closed for this function to succeed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: AddStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Adds a new style to this element.

```
virtual Acad::ErrorStatus AddStyle(  
    AcDbObjectId& styleRefId,  
    const AcDbObjectId& styleId,  
    AcMapDMAllStyleReferencesIterator& Position  
);
```

Parameters	Description
styleRefId	Output ID of the added style reference.
styleId	Input ID of the style.
Position	Input position at which to add the style. See also <a href="#">AcMapDMAllStyleReferencesIterator</a>

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The style object and the owning AcMapDMMap object must be closed for this function to succeed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: DisplayName Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Gets an alternative, constant name for this type of element in certain situations where the instance name may be unavailable.

```
virtual const ACHAR* DisplayName() const;
```

Returns

Returns a const ACHAR\* alternative name for certain display purposes.

Remarks

Derived classes should return a constant, generally descriptive name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: EvaluateExpressionValues Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: EvaluateExpressionValues Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement::ExcludeEntitiesFromStylization Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

This function may be called by derived classes during their AcquireEntities() implementation, if the Element wishes to explicitly exclude certain entities from being stylized under all circumstances.

```
Acad::ErrorStatus ExcludeEntitiesFromStylization(  
    const AcDbObjectIdArray& ids  
);
```

Parameters	Description
ids	This is the AcDbObjectIdArray of the entities that the Element wishes to exclude from stylization.

## Returns

Returns Acad::eOk if successful, otherwise, returns a different error code.

## Remarks

This is rare, but may be appropriate if the Element (for example) creates temporary entities meant for a purpose other than stylization.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)  
AcMapDMElement:: GetAcquisitionCriteria Method  
[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves a string representation of the acquisition criteria.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters	Description
pszString	Output string representation of the query.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: GetTitle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the name of this element. See also SetTitle().

```
Acad::ErrorStatus GetTitle(  
    const ACHAR*& kpszTitle  
) const;
```

Parameters	Description
kpszTitle	Output element name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: IsTopElement Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines if this element is the top element in the map.

**bool** IsTopElement() **const**;

Returns

Returns true if the element is the top element; false on any error or if the element is not the top element.

Remarks

The element must exist in the map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: MoveStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Moves a style within an element.

```
Acad::ErrorStatus MoveStyle(  
    const AcDbObjectId& styleRefId,  
    AcMapDMAllStyleReferencesIterator& newPosition  
);
```

Parameters	Description
styleRefId	Input ID of the style reference.
newPosition	Input new position to move to. See also <a href="#">AcMapDMAllStyleReferencesIterator</a>

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidOwnerObject if the specified style reference belongs to another element. Returns Acad::eInvalidIndex if newPosition is invalid. Returns Acad::eUnrecoverableErrors if a fatal error occurred.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: RemoveEntitiesFromStylizationExclusion Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

This function may be called by derived classes which used the excludeEntityFromStylization() method.

```
Acad::ErrorStatus RemoveEntitiesFromStylizationExclusion(  
    const AcDbObjectIdArray& ids  
);
```

Parameters

Description

ids

This is the AcDbObjectIdArray of the entities that the Element no longer wishes to exclude from stylization.

Returns

Returns Acad::eOk if successful, otherwise, returns a different error code.

Remarks

When the Element is done excluding the Entity from stylization, it can call this method to alert the system that this particular entity no longer requires specialized handling.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: RemoveEntityFromAcquisition Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Removes the indicated entityId from the Element's acquisition set.

```
virtual Acad::ErrorStatus RemoveEntityFromAcquisition(  
    AcDbObjectId entityId  
);
```

Parameters

Description

entityId

Input AcDbObjectId of entity to remove from Acquisition Set.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: RemoveStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Removes a style from this element.

```
Acad::ErrorStatus RemoveStyle(  
    AcMapDMAAllStyleReferencesIterator& Position  
);
```

Parameters

Description

Position

Input position of the style to remove. See also  
[AcMapDMAAllStyleReferencesIterator](#)

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: Render Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Stylize and Render layer contents to the specified renderer.

```
virtual Acad::ErrorStatus Render(  
    Renderer & renderer  
);
```

### Parameters

### Description

renderer

The Renderer onto which the element will render all layer graphics.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Remarks

This is used in various places including creating a bitmap representing the layer.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: SetCustomPreviewEntityName Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the name for kPreviewBlock and kPreviewCustom preview entity types.  
See also [GetCustomPreviewEntityName\(\)](#).

```
virtual Acad::ErrorStatus SetCustomPreviewEntityName(  
    const ACHAR* pName  
);
```

Parameters	Description
pName	Input name of the block.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement::SetTitle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the name of this element. See also GetTitle().

```
Acad::ErrorStatus SetTitle(  
    const ACHAR* kpszTitle  
);
```

Parameters	Description
kpszTitle	Input element name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMEntityStyle Class

[Classes](#)

Encapsulates properties that can be applied to entities.

```
class AcMapDMEntityStyle : public AcMapDMStyle;
```

File

DmEntityStyle.h

☐ Methods



[~AcMapDMEntityStyle](#) Destroys an instance of this class.



[AcMapDMEntityStyle](#) Constructs an instance of this class.

[Apply](#) Applies the style to an entity.

[ClearColor](#) Clears the color. See also [IsColorCleared\(\)](#).

[ClearLinetype](#) Clears the linetype. See also [IsLinetypeCleared\(\)](#).

[ClearLinetypeScale](#) Clears the linetype scale. See also [IsLinetypeScaleCleared\(\)](#).

[ClearLineWeight](#) Clears the linewidth. See also [IsLineWeightCleared\(\)](#).

[ClearPlotstyleName](#) Clears the plotstyle. See also [IsPlotstyleNameCleared\(\)](#).

[clone](#) Clones the object. See also [clone\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also [copyFrom\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Dismiss](#) Clears the stylization of a topology.

[Dismiss](#) Clears the stylization on an entity.

<a href="#"><u>dwgInFields</u></a>	Lets this object read its data. See also <code>dwgInFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>dwgOutFields</u></a>	Lets this object write its data. See also <code>dwgOutFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>dxfinFields</u></a>	Lets this object read its data. See also <code>dxfinFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>dxfoutFields</u></a>	Lets this object write its data. See also <code>dxfoutFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>GetColor</u></a>	Retrieves the color. See also <code>SetColor()</code> .
<a href="#"><u>GetLinetype</u></a>	Retrieves the linetype id. See also <code>SetLinetype()</code> .
<a href="#"><u>GetLinetype</u></a>	Retrieves the linetype name. See also <code>SetLinetype()</code> .
<a href="#"><u>GetLinetypeScale</u></a>	Retrieves the linetype scale. See also <code>SetLinetypeScale()</code> .
<a href="#"><u>GetLineWeight</u></a>	Retrieves the lineweight. See also <code>SetLineWeight()</code> .
<a href="#"><u>GetPlotstyleName</u></a>	Retrieves the plotstyle. See also <code>SetPlotstyleName()</code> .
<a href="#"><u>IsColorCleared</u></a>	Determines whether the color is cleared. See also <code>ClearColor()</code> .
<a href="#"><u>IsLinetypeCleared</u></a>	Determines whether the linetype is cleared. See also <code>ClearLinetype()</code> .
<a href="#"><u>IsLinetypeScaleCleared</u></a>	Determines whether the linetype scale is cleared. See also <code>ClearLinetypeScale()</code> .
<a href="#"><u>IsLineWeightCleared</u></a>	Determines whether the lineweight is cleared. See also <code>ClearLineWeight()</code> .
<a href="#"><u>IsPlotstyleNameCleared</u></a>	Determines whether the plotstyle is cleared. See also <code>ClearPlotstyleName()</code> .
<a href="#"><u>Preview</u></a>	Applies style traits to yield a preview of the specified entity.

<a href="#">SetColor</a>	Sets the color. See also GetColor().
<a href="#">SetLinetype</a>	Sets the linetype. See also GetLinetype().
<a href="#">SetLinetype</a>	Sets the linetype. See also GetLinetype().
<a href="#">SetLinetypeScale</a>	Sets the linetype scale. See also GetLinetypeScale().
<a href="#">SetLineWeight</a>	Sets the lineweight. See also GetLineWeight().
<a href="#">SetPlotstyleName</a>	Sets the plotstyle. See also GetPlotstyleName().
<a href="#">UnApply</a>	Removes the style from a specified entity.
<a href="#">Update</a>	Retrieves and stylizes a topology.
<a href="#">Update</a>	Retrieves and stylizes an entity.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle::Dismiss Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the stylization of a topology.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: GetLinetype Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the linetype name. See also SetLinetype().

```
Acad::ErrorStatus GetLinetype(  
    const ACHAR*& pLinetypeName  
) const;
```

Parameters	Description
pLinetypeName	Output linetype name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: GetPlotstyleName Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the plotstyle. See also SetPlotstyleName().

```
Acad::ErrorStatus GetPlotstyleName(  
    const ACHAR*& pPlotStyleName  
) const;
```

Parameters	Description
pPlotStyleName	Output plotstyle name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: Preview Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: SetLinetype Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the linetype. See also GetLinetype().

```
Acad::ErrorStatus SetLinetype(  
    const ACHAR* pLinetypeName  
);
```

Parameters	Description
pLinetypeName	Input linetype name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: SetPlotstyleName Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the plotstyle. See also [GetPlotstyleName\(\)](#).

```
Acad::ErrorStatus SetPlotstyleName(  
    const ACHAR* pPlotStyleName  
);
```

Parameters	Description
pPlotStyleName	Input plotstyle name.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: Update Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves and stylizes a topology.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoId	Input topology ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMFeatureDataSourceDescriptor Class

[Classes](#)

Represents a feature-class data source.

```
class AcMapDMFeatureDataSourceDescriptor : public AcMapDMDataSourceDescriptor  
File
```

DmFeatureElement.h

☐ Methods



[~AcMapDMFeatureDataSourceDescriptor](#) Destroys an instance of this class.



[AcMapDMFeatureDataSourceDescriptor](#) Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[SetAcquisitionStatement](#)

Sets the query string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureDataSourceDescriptor Class](#),

[AcMapDMFeatureDataSourceDescriptor Class](#)

AcMapDMFeatureDataSourceDescriptor:: GetAcquisitionStatement Method

[AcMapDMFeatureDataSourceDescriptor Class](#) |

[AcMapDMFeatureDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureDataSourceDescriptor Class](#),

[AcMapDMFeatureDataSourceDescriptor Class](#)

AcMapDMFeatureDataSourceDescriptor:: SetAcquisitionStatement Method

[AcMapDMFeatureDataSourceDescriptor Class](#) |

[AcMapDMFeatureDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMFeatureElement Class

[Classes](#)

Represents a feature-class element.

```
class AcMapDMFeatureElement : public AcMapDMCurrentDwgQueryElement;  
File
```

DmFeatureElement.h

☐ Methods

⇒ [~AcMapDMFeatureElement](#)

Destroys an instance of this class.

⇒ [AcMapDMFeatureElement](#)

Constructs an instance of this class.

[AcquireEntities](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutfields](#)

Lets this object write its data. See also `dxfoutfields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[OnMapProjectInitialized](#)

Invoked when an AutoCAD Map project is initialized.

[SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
AcMapDMFeatureElement:: EvaluateExpressionValues Method  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
[AcMapDMFeatureElement:: EvaluateExpressionValues Method](#)  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
[AcMapDMFeatureElement:: GetAcquisitionCriteria Method](#)  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters	Description
pszString	Output pointer to a data-source descriptor, as a string representation.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMGroup Class

[Classes](#)

Represents a group of display-management items (AcMapDMItem).

```
class AcMapDMGroup : public AcMapDMItem;
```

File

DmGroup.h

☐ Methods

⇒ [~AcMapDMGroup](#)

Destroys an instance of this class.

⇒ [AcMapDMGroup](#)

Constructs an instance of this class.

[AddItem](#)

Adds a new item to this group.

[AddItem](#)

Adds a new item to this group.

[audit](#)

This function is called by AutoCAD when the AUDIT command is executed.

[DismissStylization](#)

Clears the current stylization.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it

	directly.
<a href="#">dxfoutFields</a>	Lets this object write its data. See also dxfoutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">erased</a>	Lets this object listen to erase-notifications from items that it owns. See also erased() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">HasChildrenInSpecifiedVisibleState</a>	Determines whether children are visible or invisible at a specified scale.
<a href="#">IsEmpty</a>	Determines whether this group is empty.
<a href="#">IsVisible</a>	Determines whether this group is visible at the specified scale. See also setVisible()and <a href="#">HasChildrenInSpecifiedVisibleState()</a> .
<a href="#">MoveItem</a>	Moves an item to a different position in this group.
<a href="#">NewAllItemsIterator</a>	Retrieves a new iterator over the contents of this group.
<a href="#">OnMapCSChanged</a>	Invoked when an AutoCAD Map changes project coordinate system.
<a href="#">OnMapProjectInitialized</a>	Invoked when an AutoCAD Map project is initialized.
<a href="#">OnObjectAppended</a>	Invoked when an object is appended to the database.
<a href="#">RemoveItem</a>	Removes an item from this group.
<a href="#">RemoveItem</a>	Removes an item from this group.
<a href="#">SelectElementsWithStyleApplied</a>	Retrieves the IDs of the elements that have references to the specified style at the specified scale.
	Sets the group visibility at the specified

[SetVisible](#)

scale. See also [IsVisible\(\)](#) and [HasChildrenInSpecifiedVisibleState\(\)](#).

[subErase](#)

Invoked from within [erase\(\)](#) before the erase actually occurs. See also [subErase\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[UpdateStylization](#)

Stylizes the current or updated selection at the current scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: AddItem Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Adds a new item to this group.

```
virtual Acad::ErrorStatus AddItem(  
    AcDbObjectId& Id,  
    AcMapDMItem* pItem,  
    AcMapDMAllItemsIterator* pPosition = NULL  
);
```

Parameters	Description
Id	Output ID of the added item.
pItem	Input AcMapDMItemto add.
pPosition	Input position at which to add the item. The default value, NULL, adds the item to the end of the group. See also <a href="#">AcMapDMAllItemsIterator</a> .

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: AddItem Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Adds a new item to this group.

```
virtual Acad::ErrorStatus AddItem(  
    const AcDbObjectId& Id,  
    AcMapDMAllItemsIterator* pPosition = NULL  
);
```

Parameters	Description
Id	Input ID of the item to add.
pPosition	Input position at which to add the item. The default value, NULL, adds the item to the end of the group. See also <a href="#">AcMapDMAllItemsIterator</a> .

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: MoveItem Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Moves an item to a different position in this group.

```
virtual Acad::ErrorStatus MoveItem(  
    const AcDbObjectId& Id,  
    AcMapDMAllItemsIterator& Position  
);
```

Parameters

Description

Id

Input ID of the item to move.

Position

Input position to move the item to. See also  
[AcMapDMAllItemsIterator](#).

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidIndex if the iterator position is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: OnMapCSChanged Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Invoked when an AutoCAD Map changes project coordinate system.

```
virtual void OnMapCSChanged(  
    const ACHAR * oldCS,  
    const ACHAR * newCS  
);
```

Parameters	Description
oldCS	Old CS code
newCS	New CS code

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

[AcMapDMGroup:: RemoveItem Method](#)

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Removes an item from this group.

```
virtual Acad::ErrorStatus RemoveItem(  
    AcMapDMAllItemsIterator& Position  
);
```

Parameters

Description

Position

Input position of the item to remove. See also [AcMapDMAllItemsIterator](#).

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidIndex if the iterator position is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMItem Class

[Classes](#)

Base class of all display-management items.

```
class AcMapDMItem : public AcDbObject;  
File
```

DmDisplayItem.h

☐ Enumerations



[ELegendDetailLevel](#)

Enumerates the levels of legend detail.

☐ Methods

◆ [~AcMapDMItem](#)

Destroys an instance of this class.

◆ [AcMapDMItem](#)

Constructs an instance of this class.

[audit](#)

This function is called by AutoCAD when the AUDIT command is executed.

[DismissStylization](#)

Clears the current stylization.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed;

<a href="#">GetName</a>	it is unlikely that you will need to call it directly. Retrieves the name of this item. See also <a href="#">SetName()</a> .
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">IsVisible</a>	Determines whether this item is visible at the specified scale. See also <a href="#">SetVisible()</a> .
<a href="#">LegendDetailLevel</a>	Retrieves the level of legend detail of this item. See also <a href="#">SetLegendDetailLevel()</a> .
<a href="#">MapId</a>	Retrieves the map ID.
<a href="#">MapProject</a>	Retrieves the AutoCAD Map project.
<a href="#">OnMapCSChanged</a>	Invoked when an AutoCAD Map changes project coordinate system.
<a href="#">OnMapProjectInitialized</a>	Invoked when an AutoCAD Map project is initialized.
<a href="#">OnObjectAppended</a>	Invoked when an object is appended to the database.
<a href="#">SetLegendDetailLevel</a>	Sets the level of legend detail of this item. See also <a href="#">LegendDetailLevel()</a> .
<a href="#">SetName</a>	Sets the name of this item. See also <a href="#">GetName()</a> .
<a href="#">SetVisible</a>	Sets the item visibility at the specified scale. See also <a href="#">IsVisible()</a> .
<a href="#">subClose</a>	Invoked from within <a href="#">close()</a> before the close actually occurs. The default implementation of this function returns <code>Acad::eOk</code> . See also <a href="#">subClose()</a> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">subErase</a>	Invoked from within <a href="#">erase()</a> before the erase actually occurs. See also <a href="#">subErase()</a> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">UpdateStylization</a>	Stylizes the current or updated selection at the current scale.
<a href="#">wblockClone</a>	Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object <code>pOwnerObject</code> . See also <a href="#">wblockClone()</a> in the AutoCAD ObjectARX

Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)  
AcMapDMItem:: OnMapCSChanged Method  
[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Invoked when an AutoCAD Map changes project coordinate system.

```
virtual void OnMapCSChanged(  
    const ACHAR * oldCS,  
    const ACHAR * newCS  
);
```

Parameters	Description
oldCS	Old CS code
newCS	New CS code

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem::SetName Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Sets the name of this item. See also GetName().

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input item name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMLayerDataSourceDescriptor Class

[Classes](#)

Represents a layer data source.

```
class AcMapDMLayerDataSourceDescriptor : public AcMapDMDataSourceDes  
File
```

DmLayerElement.h

☐ Methods



[~AcMapDMLayerDataSourceDescriptor](#) Destroys an instance of this class.



[AcMapDMLayerDataSourceDescriptor](#) Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[SetAcquisitionStatement](#)

Sets the query string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerDataSourceDescriptor Class](#),

[AcMapDMLayerDataSourceDescriptor Class](#)

AcMapDMLayerDataSourceDescriptor:: GetAcquisitionStatement Method

[AcMapDMLayerDataSourceDescriptor Class](#) |

[AcMapDMLayerDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerDataSourceDescriptor Class](#),

[AcMapDMLayerDataSourceDescriptor Class](#)

AcMapDMLayerDataSourceDescriptor:: SetAcquisitionStatement Method

[AcMapDMLayerDataSourceDescriptor Class](#) |

[AcMapDMLayerDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMLayerElement Class

[Classes](#)

Represents a layer element.

```
class AcMapDMLayerElement : public AcMapDMCurrentDwgQueryElement;  
File
```

DmLayerElement.h

☐ Methods

⇒ [~AcMapDMLayerElement](#)

Destroys an instance of this class.

⇒ [AcMapDMLayerElement](#)

Constructs an instance of this class.

[AcquireEntities](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutfields](#)

Lets this object write its data. See also [dxfoutfields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[OnMapProjectInitialized](#)

Invoked when an AutoCAD Map project is initialized.

[SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: EvaluateExpressionValues Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: EvaluateExpressionValues Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: GetAcquisitionCriteria Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters	Description
pszString	Output pointer to the data-source descriptor, as a string representation.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMLegend Class

[Classes](#)

Represents a legend object. A legend has the following characteristics: A legend is a table. A legend will display the active (checked) Themes, Elements, and Groups as in the display-manager tree. The order of items in the legend matches the order of items in the display-manager tree. Each "space" (modelspace, layout1, layoutN, and so on) can have zero or one legend, which is built from the current Map and Scale when a legend is inserted or updated. A legend object does not store information on which layout or map it is associated with. .

```
class AcMapDMLegend : public AcDbTable;
```

File

DmLegend.h

☐ Methods

◆ [~AcMapDMLegend](#) Destroys an instance of this class.

◆ [AcMapDMLegend](#) Constructs an instance of this class.

[DismissTableContents](#) Clears the contents of the legend.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#) Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed;

it is unlikely that you will need to call it directly.

[Implementation](#)

Returns the implementation object.

[UpdateTableContents](#)

Clears the contents of the legend and updates it with current data from the display manager.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMMMap Class

[Classes](#)

Represents the Display Manager map, the root object from which all other Display Manager objects can be obtained by using an iterator.

```
class AcMapDMMMap : public AcMapDMGroup;
```

File

DmMap.h

☐ Enumerations

📄 [DOMode](#) This is record AcMapDMMMap::DOMode.

☐ Methods

🔗 [~AcMapDMMMap](#)

Destroys an instance of this class.

🔗 [AcMapDMMMap](#)

Constructs an instance of this class.

[AddElement](#)

Adds an element to this map.

[AddReactor](#)

Adds a reactor to this map.

[AddScaleThreshold](#)

Adds a scale threshold to this map - this function is not yet supported in AutoCAD Map.

[audit](#)

This function is called by AutoCAD when the AUDIT command is executed.

[CopyScaleThreshold](#)

Copies a scale threshold to this map - this function is not yet supported in AutoCAD Map.

[CreateLegend](#)

Creates a new legend for the specified layout.

[deepCloneObject](#)

Clones this object and all the objects that it refers to.

[deepCloneObjects](#)

Clones this object and all the objects that it refers to.

[DeleteLegend](#)

Deletes the contents of the legend of the specified layout.

[DismissStylization](#)

Clears the current stylization.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[erased](#)

Lets this object listen to erase-notifications from items that it owns. See also `erased()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Get3dGridPercent](#)

Retrieves the current 3d grid percent of this map exactly as last input to `Set3dGridPercent()`.

[GetAllDrawOrderItemsIterator](#)

Retrieves the draw-order iterator. The first function call iterates elements in the order that they are arranged in the map tree. This iterator may return Items that are in a proxy state, unless the `bSkipProxies` argument is passed as true. See also `IsDrawOrderDefined()`.

<a href="#"><u>GetCurrentScaleThreshold</u></a>	Retrieves the current scale threshold of this map, as calculated from the last input to SetCurrentScale().
<a href="#"><u>GetCurrentUserScale</u></a>	Retrieves the current scale of this map exactly as last input to SetCurrentScale().
<a href="#"><u>GetDOMode</u></a>	Gets the draw order mode currently in use for the map.
<a href="#"><u>GetExaggeration</u></a>	Retrieves the current exaggeration of this map exactly as last input to SetExaggeration().
<a href="#"><u>GetExtent</u></a>	Retrieves the current extent of this map exactly as last input to SetExtent().
<a href="#"><u>GetLegend</u></a>	Retrieves the legend of the specified layout.
<a href="#"><u>GetLegendId</u></a>	Retrieves the legend of the specified layout.
<a href="#"><u>GetLineStyleScale</u></a>	Retrieves the linetype scale applied to the stylization. The linetype scale, a positive value, is assigned to the AutoCAD system variable LTSCALE at update time. See also SetLineStyleScale().
<a href="#"><u>GetLinkedFileName</u></a>	Retrieves the linked output filename. The linked output file contains the stylized objects of this map. By default, the current scale is used in output. See also SetLinkedFileName().
<a href="#"><u>GetName</u></a>	Retrieves the name of this map. See also SetName().
<a href="#"><u>GetScaleThreshold</u></a>	Retrieves the scale threshold at a specified index.
<a href="#"><u>GetSun</u></a>	Get sun vector setting for current map.
<a href="#"><u>GetTableStyle</u></a>	Retrieves the table style. See also SetTableStyle().
<a href="#"><u>GetThresholdFor</u></a>	Retrieves an upper-bound map threshold scale for the specified scale.
<a href="#"><u>GetThumbnailDimensions</u></a>	Retrieves the thumbnail width and height factors. These width and height factors are multiplied by the text height in the legend to

<a href="#">GetThumbnailFraming</a>	determine the size of the thumbnail width and height. See also <a href="#">SetThumbnailDimensions()</a> . Determines whether a box is drawn around the thumbnail graphic. See also <a href="#">SetThumbnailFraming()</a> .
<a href="#">InfinityThreshold</a>	Returns the value used as the Infinity threshold scale in the Map. Determines whether the draw order is defined. Draw order is defined on demand when <a href="#">GetDrawOrderIterator()</a> is first called. Draw-order functionality is available in Display Manager user interface in the list view.
<a href="#">IsDrawOrderDefined</a>	Reports whether the input value is equal to the value used as the Infinity threshold scale in the Map.
<a href="#">IsInfintyThreshold</a>	Determines whether linked-file capability is enabled.
<a href="#">IsLinkToFileEnabled</a>	Determines whether this map is stylized.
<a href="#">IsStylized</a>	Determines whether this map is visible at the specified scale. See also <a href="#">SetVisible()</a> .
<a href="#">IsVisible</a>	Retrieves the ID of this map.
<a href="#">MapId</a>	Changes the scale threshold value of an existing scale threshold.
<a href="#">ModifyScaleThreshold</a>	Changes the position of an element within this map.
<a href="#">MoveElement</a>	Counts the number of scale thresholds defined in this map.
<a href="#">NumScaleThresholds</a>	Invoked when an AutoCAD Map project is initialized.
<a href="#">OnMapProjectInitialized</a>	Invoked when an object is appended to the database.
<a href="#">OnObjectAppended</a>	Makes the system to refresh the contents of map when the given style is modified.
<a href="#">OnStyleModified</a>	Removes an element from this map.
<a href="#">RemoveElement</a>	Removes the extent of this map.
<a href="#">RemoveExtent</a>	

<a href="#">RemoveReactor</a>	Removes a reactor from this map.
<a href="#">RemoveScaleThreshold</a>	Removes a scale threshold from this map.
<a href="#">RemoveScaleThreshold</a>	Removes a scale threshold from this map.
<a href="#">Requery</a>	Requeries objects belonging to the map.
	Resumes fixing of the draw order of entities. The draw order is fixed during this call. Any change that affects the draw order after this method is called until the next
<a href="#">ResumeFixingDO</a>	SuspendFixingDO method will be reflected immediately. This method <b>must</b> be used in conjunction with SuspendFixingDO() and these two methods must always be paired.
	Saves the current database in its current state of stylization at the current scale to the filename specified in a previous call to <a href="#">AcMapDMMap::SetLinkedFileName()</a> . The linked file contains a stylized map that can be plotted or published to DWF. Clicking the Update button in the AutoCAD Map user interface invokes SaveLinkedFile().
<a href="#">SaveLinkedFile</a>	SaveLinkedFile() does not call <a href="#">AcMapDMMap::UpdateStylization()</a> ; it is your application's responsibility to prepare the stylized model. SaveLinkedFile() calls AcMapDMMap::GetCurrentScale(), followed by <a href="#">AcMapDMMap::GetLinkedFileName()</a> , with the current scale to obtain the target filename.
	Retrieves elements that reference the specified style at the specified scale threshold.
<a href="#">SelectElementsWithStyleApplied</a>	
<a href="#">Set3dGridPercent</a>	Sets the 3d grid percent of this map.
	Sets the current scale of this map, affects BOTH the arbitrary scale and the classic Threshold.
<a href="#">SetCurrentScale</a>	Sets the current scale threshold of this map at

<a href="#"><u>SetCurrentScaleThreshold</u></a>	the specified index.
<a href="#"><u>SetDOMode</u></a>	Sets the draw order mode for the map.
<a href="#"><u>SetExaggeration</u></a>	Sets the current exaggeration of this map.
<a href="#"><u>SetExtent</u></a>	Sets the extent of this map.
<a href="#"><u>SetLineTypeScale</u></a>	Sets the linetype scale for a specified scale. See also <a href="#"><u>GetLineTypeScale()</u></a> .
<a href="#"><u>SetLinkedFileName</u></a>	Sets the linked output filename. The linked output file contains the stylized objects of this map. By default, current scale is used in output. See also <a href="#"><u>GetLinkedFileName()</u></a> .
<a href="#"><u>SetLinkToFileEnabled</u></a>	Sets the status of the linked file capability.
<a href="#"><u>SetName</u></a>	Sets the name of this map. See also <a href="#"><u>GetName()</u></a> .
<a href="#"><u>SetSun</u></a>	Sets sun vector setting for current map.
<a href="#"><u>SetTableStyle</u></a>	Sets the table style. The table style is used for the legend. See also <a href="#"><u>GetTableStyle()</u></a> .
<a href="#"><u>SetThumbnailDimensions</u></a>	Sets the thumbnail width and height factors. These width and height factors are multiplied by the text height in the legend to determine the size of the thumbnail width and height. See also <a href="#"><u>GetThumbnailDimensions()</u></a> .
<a href="#"><u>SetThumbnailFraming</u></a>	Sets whether a box is drawn around the thumbnail graphic. See also <a href="#"><u>GetThumbnailFraming()</u></a> .
<a href="#"><u>SetVisible</u></a>	Sets the map visibility at the specified scale. See also <a href="#"><u>IsVisible()</u></a> .
<a href="#"><u>subErase</u></a>	Invoked from within <a href="#"><u>erase()</u></a> before the erase actually occurs. See also <a href="#"><u>subErase()</u></a> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>SuspendFixingDO</u></a>	Suspends fixing of the draw order of entities.
<a href="#"><u>UpdateStylization</u></a>	Stylizes the current or updated selection at the current scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::AddElement Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Adds an element to this map.

```
Acad::ErrorStatus AddElement(  
    AcDbObjectId& Id,  
    AcMapDMElement* pElement,  
    AcMapDMAAllDrawOrderItemsIterator& Position  
);
```

Parameters	Description
Id	Output ID of the map.
pElement	Input AcMapDMElementto add.
Position	Input position at which to add the element. See also <a href="#">AcMapDMAAllDrawOrderItemsIterator</a> .

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: AddScaleThreshold Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Adds a scale threshold to this map - this function is not yet supported in AutoCAD Map.

```
Acad::ErrorStatus AddScaleThreshold(  
    double dNewScale,  
    double dSourceScale = 0.  
);
```

Parameters	Description
dNewScale	Input new scale threshold.
dSourceScale	Input source scale threshold. The default value is zero.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For every element, this function adds a new scale and associates it with the style set and with the source scale where dSourceScale is defined on the map. If dSourceScale is zero (the default), a new empty style set is created.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: CopyScaleThreshold Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Copies a scale threshold to this map - this function is not yet supported in AutoCAD Map.

```
Acad::ErrorStatus CopyScaleThreshold(  
    double dNewScale,  
    double dSourceScale  
);
```

Parameters	Description
dNewScale	Input new scale threshold.
dSourceScale	Input source scale threshold. The default value is zero.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For every element, this function copies a new scale and associates it with the style set, and associates with the source scale where dSourceScale is defined on the map. If dSourceScale is zero (the default), a new empty style set is created.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: Get3dGridPercent Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the current 3d grid percent of this map exactly as last input to Set3dGridPercent().

```
int Get3dGridPercent() const;
```

Returns

Returns the 3d grid percent.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetCurrentScaleThreshold Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the current scale threshold of this map, as calculated from the last input to SetCurrentScale().

**double** GetCurrentScaleThreshold() **const**;

Returns

Returns the scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::GetCurrentUserScale Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the current scale of this map exactly as last input to SetCurrentScale().

```
double GetCurrentUserScale() const;
```

Returns

Returns the scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: GetExaggeration Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the current exaggeration of this map exactly as last input to SetExaggeration().

```
double GetExaggeration() const;
```

Returns

Returns the exaggeration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: GetExtent Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the current extent of this map exactly as last input to SetExtent().

```
Acad::ErrorStatus GetExtent(  
    double& dMinX,  
    double& dMinY,  
    double& dMaxX,  
    double& dMaxY  
) const;
```

Parameters	Description
dMinX	Output the extent min x value.
dMinY	Output the extent min y value.
dMaxX	Output the extent max x value. Output the extent max Y value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetLinkedFileName Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the linked output filename. The linked output file contains the stylized objects of this map. By default, the current scale is used in output. See also SetLinkedFileName().

```
Acad::ErrorStatus GetLinkedFileName(  
    const ACHAR*& kpszFileName,  
    double dScale = 0.  
) const;
```

Parameters	Description
kpszFileName	Output linked filename.
dScale	Input threshold scale value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: GetScaleThreshold Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the scale threshold at a specified index.

```
Acad::ErrorStatus GetScaleThreshold(  
    double& dScale,  
    size_t iIndex  
) const;
```

Parameters	Description
dScale	Output scale threshold.
iIndex	Input index of the desired scale threshold.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetSun Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Get sun vector setting for current map.

```
Acad::ErrorStatus GetSun(  
    double& dAzimuth,  
    double& dAltitude  
) const;
```

Parameters	Description
dAzimuth	Azimuth value of sun vector
dAltitude	Altitude value of sun vector

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: GetThresholdFor Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves an upper-bound map threshold scale for the specified scale.

```
double GetThresholdFor(  
    double dDisplayScale  
) const;
```

Parameters	Description
dDisplayScale	Input display scale.

Returns

Returns the map scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::InfinityThreshold Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Returns the value used as the Infinity threshold scale in the Map.

```
static double InfinityThreshold();
```

Returns

Returns the value used as the Infinity threshold scale in the Map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: IsInfintyThreshold Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Reports whether the input value is equal to the value used as the Infinity threshold scale in the Map.

```
static bool IsInfintyThreshold(  
    double dScale  
);
```

Parameters	Description
dScale	Input scale to compare to DMMMap's "infinity" threshold.

Returns

Returns true if the input value is the Infinty threshold, otherwise returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap::ModifyScaleThreshold Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Changes the scale threshold value of an existing scale threshold.

```
Acad::ErrorStatus ModifyScaleThreshold(  
    double dOldScale,  
    double dNewScale  
);
```

Parameters	Description
dOldScale	Input old scale threshold.
dNewScale	Input new scale threshold.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::MoveElement Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Changes the position of an element within this map.

```
Acad::ErrorStatus MoveElement(  
    const AcDbObjectId& Id,  
    AcMapDMAllDrawOrderItemsIterator& Position  
);
```

Parameters	Description
Id	Input ID of the element to move.
Position	Input position to move the element to. See also <a href="#">AcMapDMAllDrawOrderItemsIterator</a> .

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidIndex if the iterator position is invalid.

Remarks

Moving an element affects the draw order of entities that are part of element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: NumScaleThresholds Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Counts the number of scale thresholds defined in this map.

```
size_t NumScaleThresholds() const;
```

Returns

Returns the number of scales.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap:: RemoveElement Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Removes an element from this map.

```
Acad::ErrorStatus RemoveElement(  
    AcMapDMA11DrawOrderItemsIterator& Position  
);
```

Parameters

Description

Position

Input position of the element to remove. See also [AcMapDMA11DrawOrderItemsIterator](#).

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidIndex if the iterator position is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: RemoveExtent Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Removes the extent of this map.

```
Acad::ErrorStatus RemoveExtent();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::RemoveScaleThreshold Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Removes a scale threshold from this map.

```
Acad::ErrorStatus RemoveScaleThreshold(  
    double dScale  
);
```

Parameters

Description

dScale

Input value of the scale threshold to remove.

Returns

Returns Acad::eOk if successful. Returns Acad::eInvalidInput if the scale has not been defined. Returns Acad::eNotApplicable if the scale is DBL\_MAX.

Remarks

If the scale is the current scale, this function dismisses stylization.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::RemoveScaleThreshold Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Removes a scale threshold from this map.

```
Acad::ErrorStatus RemoveScaleThreshold(  
    size_t iIndex  
);
```

Parameters

Description

iIndex

Input index of the scale threshold to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: Set3dGridPercent Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the 3d grid percent of this map.

```
Acad::ErrorStatus Set3dGridPercent(  
    int iValue  
);
```

Parameters	Description
iValue	Input percent value. The valid value is from 10 to 100.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap:: SetCurrentScale Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Sets the current scale of this map, affects BOTH the arbitrary scale and the classic Threshold.

```
Acad::ErrorStatus SetCurrentScale(  
    double dScale,  
    bool bRegen = true  
);
```

Parameters	Description
dScale	Input arbitrary scale.
bRegen	Input whether to have Elements regen their acquired entities, regardless of whether a threshold change is occurring.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This method sets two internal values, an arbitrary "user scale" and the appropriate "map threshold scale", the latter being a threshold value corresponding to scale thresholds available for classic DM Elements. If the change in arbitrary scale does not require a threshold change, no threshold change will occur. Changing the current threshold is not allowed if stylization is on. If the Threshold will change, the caller is required to bracket this call with DismissStylization/UpdateStylization, which will requery the elements of the map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetCurrentScaleThreshold Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the current scale threshold of this map at the specified index.

```
Acad::ErrorStatus SetCurrentScaleThreshold(  
    size_t iIndex  
);
```

Parameters	Description
iIndex	Input index of the scale threshold.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Sets the current User scale to match the specified threshold. Setting the current threshold requeries the elements of the map and applies stylization if stylization is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: SetExaggeration Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the current exaggeration of this map.

```
Acad::ErrorStatus SetExaggeration(  
    double dValue  
);
```

Parameters	Description
dValue	Input exaggeration value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetExtent Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the extent of this map.

```
Acad::ErrorStatus SetExtent(  
    double dMinX,  
    double dMinY,  
    double dMaxX,  
    double dMaxY  
);
```

Parameters	Description
dMinX	Input the extent min x value.
dMinY	Input the extent min y value.
dMaxX	Input the extent max x value. Input the extent max Y value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetLinkedFileName Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the linked output filename. The linked output file contains the stylized objects of this map. By default, current scale is used in output. See also GetLinkedFileName().

```
Acad::ErrorStatus SetLinkedFileName(  
    const ACHAR* kpszFileName,  
    double dScale = 0.  
);
```

Parameters	Description
kpszFileName	Input linked filename.
dScale	Input threshold scale value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap::SetName Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the name of this map. See also `GetName()`.

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input name of the map.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetSun Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets sun vector setting for current map.

```
Acad::ErrorStatus SetSun(  
    double dAzimuth,  
    double dAltitude  
);
```

Parameters	Description
dAzimuth	Azimuth value of sun vector
dAltitude	Altitude value of sun vector

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetTableStyle Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the table style. The table style is used for the legend. See also GetTableStyle().

```
Acad::ErrorStatus SetTableStyle(  
    const ACHAR* pszTableName  
);
```

Parameters	Description
pszTableName	Input name of the table style.

Returns

Returns Acad::eOk if a table style with the specified name exists. Returns Acad::eInvalidInput if pszTableName is a NULL pointer or empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMMapIterator Class

[Classes](#)

An iterator over a collection of display-management map objects (AcMapDMMap).

```
class AcMapDMMapIterator;
```

File

DmMapManager.h

☐ Methods



[~AcMapDMMapIterator](#) Destroys an instance of this class.

[Done](#) Determines whether the iterator has reached the end of the collection.

[GetObj](#) Retrieves the current object in the iteration.

[Next](#) Advances to the next element in the iteration.

[ObjectId](#) Retrieves the ID of the current object in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMMapManager Class

[Classes](#)

Manages display-management map objects (AcMapDMMap).

```
class AcMapDMMapManager : public AcDbDictionaryWithDefault;  
File
```

DmMapManager.h

☐ Methods



[~AcMapDMMapManager](#) Destroys an instance of this class.



[AcMapDMMapManager](#) Constructs an instance of this class.

[AddDefaultMap](#) Creates a default map object.

[Append](#) Adds a map.

[CreateNewMap](#) Adds a new map with the default settings.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#) Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is

	called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">Export</a>	Exports an XML string representation of a map. See also Import().
<a href="#">GetAt</a>	Retrieves the ID of the named map.
<a href="#">GetAt</a>	Retrieves the named map.
<a href="#">GetCurrent</a>	Retrieves the ID of the current map.
<a href="#">GetCurrent</a>	Retrieves the name of the current map.
<a href="#">Has</a>	Determines whether a map with the specified ID exists.
<a href="#">Has</a>	Determines whether a named map exists.
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">Import</a>	Imports an XML string to form a map - this function is not yet implemented in AutoCAD Map. See also Export().
<a href="#">NewIterator</a>	Returns a new map iterator.
<a href="#">NumMaps</a>	Counts the number of maps.
<a href="#">Remove</a>	Removes the map with the specified ID.
<a href="#">Remove</a>	Removes the named map.
<a href="#">ResetName</a>	Renames the named map.
<a href="#">ResetName</a>	Renames the map with the specified ID.
<a href="#">SetCurrent</a>	Sets the named map to be the current map.
<a href="#">SetCurrent</a>	Sets the map with the specified ID to be the current map.
<a href="#">wblockClone</a>	Clones the specified map to the destination database. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">wblockCloneObjects</a>	Deep clones the specified objects and appends them to the specified container. The objects can come from multiple source databases, and must match the type of owner specified, but must be from a different database than the ownerId object. See also wblockCloneObjects() in the AutoCAD ObjectARX

Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Append Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Adds a map.

```
Acad::ErrorStatus Append(  
    AcDbObjectId& Id,  
    const ACHAR* pszName,  
    AcMapDMMMap* pMap  
);
```

Parameters	Description
Id	Output ID of the added map.
pszName	Input name of the map.
pMap	Input AcMapDMMMap.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Export Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Exports an XML string representation of a map. See also Import().

```
Acad::ErrorStatus Export(  
    ACHAR*& pszXMLString  
);
```

Parameters	Description
pszXMLString	Output XML string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: GetAt Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Retrieves the ID of the named map.

```
Acad::ErrorStatus GetAt(  
    AcDbObjectId& Id,  
    const ACHAR* pszName  
) const;
```

Parameters	Description
Id	Output ID of the map.
pszName	Input name of the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: GetAt Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Retrieves the named map.

```
Acad::ErrorStatus GetAt(  
    AcMapDMMMap*& pMap,  
    const ACHAR* pszName,  
    AcDb::OpenMode mode  
) const;
```

Parameters	Description
pMap	Output AcMapDMMMap.
pszName	Input name of the map.
mode	Input mode for opening the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager::GetCurrent Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Retrieves the name of the current map.

```
Acad::ErrorStatus GetCurrent(  
    const ACHAR*& pszName  
) const;
```

Parameters	Description
pszName	Output name of the current map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Has Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Determines whether a named map exists.

```
bool Has(  
    const ACHAR* pszName  
) const;
```

Parameters	Description
pszName	Input name of the map to search for.

Returns

Returns true if the map exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Import Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Imports an XML string to form a map - this function is not yet implemented in AutoCAD Map. See also Export().

```
Acad::ErrorStatus Import(  
    const ACHAR* pszXMLString  
);
```

Parameters	Description
pszXMLString	Input XML string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: Remove Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Removes the named map.

```
Acad::ErrorStatus Remove(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input name of the map to remove.

Returns

Returns Acad::eOk if successful. Returns Acad::eNotApplicable if the specified map is the current map.

## Remarks

The default map cannot be removed. If the current map is removed, the default map becomes the current map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: ResetName Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Renames the named map.

```
Acad::ErrorStatus ResetName(  
    const ACHAR* pszOldName,  
    const ACHAR* pszNewName  
);
```

Parameters	Description
pszOldName	Input name of the map to rename.
pszNewName	Input new map name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: ResetName Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Renames the map with the specified ID.

```
Acad::ErrorStatus ResetName(  
    const AcDbObjectId& mapId,  
    const ACHAR* pszNewName  
);
```

Parameters	Description
mapId	Input ID of the map to rename.
pszNewName	Input new map name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: SetCurrent Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Sets the named map to be the current map.

```
Acad::ErrorStatus SetCurrent(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input name of the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMMapReactor Class

[Classes](#)

Base class used to notify an application of display-management events within a display-management map. Derive custom reactors from the AcMapDMMapReactor class.

To add a reactor: Derive your class AcMapDMMyReactor from AcMapDMMapReactor. Override the virtual functions of this reactor base class. Use [AcMapDMMap::AddReactor\(\)](#) to register an instance of the reactor. To unregister a reactor, use [AcMapDMMap::RemoveReactor\(\)](#).

```
class AcMapDMMapReactor;
```

File

DMReactor.h

☐ Methods

<a href="#">CurrentScaleModified</a>	Invoked when the current scale is modified.
<a href="#">DismissStylizationBegin</a>	Invoked just before stylization is dismissed.
<a href="#">DismissStylizationCancel</a>	Invoked when a stylization dismissal is cancelled.
<a href="#">DismissStylizationEnd</a>	Invoked just after stylization is dismissed.
<a href="#">ItemAppended</a>	Invoked when a display-management item is added to the map.
<a href="#">ItemErased</a>	Invoked when a display-management item is erased from the map.
<a href="#">ItemModified</a>	Invoked when a display-management item is modified.
<a href="#">ScaleAdded</a>	Invoked when a scale is added.
<a href="#">ScaleErased</a>	Invoked when a scale is erased.
<a href="#">ScaleModified</a>	Invoked when a scale is modified.
<a href="#">StyleAppended</a>	Invoked when a style is added.
<a href="#">StyleErased</a>	Invoked when a style is erased.
<a href="#">StyleModified</a>	Invoked when a style is modified.

[StyleReferenceAppended](#) Invoked when a style reference is added.  
[StyleReferenceErased](#) Invoked when a style reference is erased.  
[StyleReferenceModified](#) Invoked when a style reference is modified.  
[UpdateStylizationBegin](#) Invoked just before stylization occurs.  
[UpdateStylizationCancel](#) Invoked when a stylization is cancelled.  
[UpdateStylizationEnd](#) Invoked just after stylization occurs.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMProjectReactor Class

[Classes](#)

Base class used to notify the an application of display-management events within an AutoCAD Map project. Derive custom reactors from the AcMapDMProjectReactor class.

To add a reactor: Derive your class AcMapDMMMyReactor from AcMapDMProjectReactor. Override the virtual functions of this reactor base class. Use [AcMapDMDisplayManagement::AddProjectReactor\(\)](#) to register an instance of the reactor. To unregister a reactor, use [AcMapDMDisplayManagement::RemoveProjectReactor\(\)](#).

```
class AcMapDMProjectReactor;
```

File

DMReactor.h

☐ Methods

<a href="#">CategoryAppended</a>	Invoked when a category is added to the style library.
<a href="#">CategoryModified</a>	Invoked when a category is modified.
<a href="#">CategoryUnAppended</a>	Invoked when a category is detached from the style library.
<a href="#">MapAppended</a>	Invoked when a new map is appended or after a reactor is attached to a project that has map(s) defined.
<a href="#">MapGoodBye</a>	Invoked when a map is unloaded.
<a href="#">MapSetCurrentBegin</a>	Invoked just before a map is set as the current map.
<a href="#">MapSetCurrentEnd</a>	Invoked just after a map is set as the current map.
<a href="#">MapSetCurrentFails</a>	Invoked when setting a map as current fails.
<a href="#">MapUnAppended</a>	Invoked when a map is detached.
<a href="#">StyleAppended</a>	Invoked when a style is added to the category.
<a href="#">StyleModified</a>	Invoked when a style is modified.
<a href="#">StyleUnAppended</a>	Invoked when a style is detached.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMQueryDataSourceDescriptor Class

[Classes](#)

Describes a base data source for querying.

```
class AcMapDMQueryDataSourceDescriptor : public AcMapDMDataSourceDes  
File
```

DmProjectQueryElement.h

☐ Methods



[~AcMapDMQueryDataSourceDescriptor](#) Destroys an instance of this class.



[AcMapDMQueryDataSourceDescriptor](#) Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[GetQuery](#)

Retrieves the query definition.

[SetAcquisitionStatement](#)

Sets the query string.

[SetQuery](#)

Sets the query definition.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor:: GetAcquisitionStatement Method

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor:: SetAcquisitionStatement Method

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMRasterDataSourceDescriptor Class

[Classes](#)

Defines a raster data source.

```
class AcMapDMRasterDataSourceDescriptor : public AcMapDMDataSourceDe  
File
```

DmRasterElement.h

☐ Methods



[~AcMapDMRasterDataSourceDescriptor](#) Destroys an instance of this class.



[AcMapDMRasterDataSourceDescriptor](#) Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[SetAcquisitionStatement](#)

Sets the query string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterDataSourceDescriptor Class](#),

[AcMapDMRasterDataSourceDescriptor Class](#)

AcMapDMRasterDataSourceDescriptor:: GetAcquisitionStatement Method

[AcMapDMRasterDataSourceDescriptor Class](#) |

[AcMapDMRasterDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterDataSourceDescriptor Class](#),

[AcMapDMRasterDataSourceDescriptor Class](#)

AcMapDMRasterDataSourceDescriptor:: SetAcquisitionStatement Method

[AcMapDMRasterDataSourceDescriptor Class](#) |

[AcMapDMRasterDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMRasterElement Class

[Classes](#)

Represents a raster element that selects raster images.

```
class AcMapDMRasterElement : public AcMapDMElement;  
File
```

DmRasterElement.h

☐ Methods

⇒ [~AcMapDMRasterElement](#)

Destroys an instance of this class.

⇒ [AcMapDMRasterElement](#)

Constructs an instance of this class.

[AcquireEntities](#)

Runs the query against the current drawing.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also

[dxfoutfields](#)

dxfoutfields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: EvaluateExpressionValues Method  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: EvaluateExpressionValues Method  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: GetAcquisitionCriteria Method  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters	Description
pszString	Output pointer to the data-source descriptor, as a string representation.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMRasterStyle Class

[Classes](#)

Encapsulates raster-image-related properties.

```
class AcMapDMRasterStyle : public AcMapDMStyle;
```

File

DmRasterStyle.h

☐ Methods



[~AcMapDMRasterStyle](#) Destroys an instance of this class.



[AcMapDMRasterStyle](#) Constructs an instance of this class.

[Apply](#) Applies a style to a raster.

[ClearBrightness](#) Clears image brightness. See also [IsBrightnessCleared\(\)](#).

[ClearContrast](#) Clears image contrast. See also [IsContrastCleared\(\)](#).

[ClearFade](#) Clears image fade. See also [IsFadeCleared\(\)](#).

[ClearTransparency](#) Clears image transparency. See also [IsTransparencyCleared\(\)](#).

[clone](#) Clones the object. See also [clone\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also [copyFrom\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[DeleteCookie](#) Deletes the style's cookie during database destruction.

[Dismiss](#) Clears the stylization of a raster.

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This

<a href="#">dwgInFields</a>	overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dwgOutFields</a>	Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dxfInFields</a>	Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dxfOutFields</a>	Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">Enable</a>	Enables or disables the style of a raster.
<a href="#">GetBrightness</a>	Retrieves image brightness. See also SetBrightness().
<a href="#">GetContrast</a>	Retrieves image contrast. See also SetContrast().
<a href="#">GetFade</a>	Retrieves image fade. See also SetFade().
<a href="#">GetTransparency</a>	Retrieves image transparency. See also SetTransparency().
<a href="#">IsBrightnessCleared</a>	Determines whether image brightness is cleared. See also ClearBrightness().
<a href="#">IsContrastCleared</a>	Determines whether image contrast is cleared. See also ClearContrast().
<a href="#">IsFadeCleared</a>	Determines whether image fade is cleared. See also ClearFade().
<a href="#">IsTransparencyCleared</a>	Determines whether image transparency is cleared. See also ClearTransparency().
<a href="#">Preview</a>	Applies style traits to yield a preview of the specified raster.
<a href="#">SetBrightness</a>	Sets image brightness. See also GetBrightness().
<a href="#">SetContrast</a>	Sets image contrast. See also GetContrast().
<a href="#">SetFade</a>	Sets image fade. See also GetFade().
<a href="#">SetTransparency</a>	Sets image transparency. See also GetTransparency().

[UnApply](#)

Removes the style from a raster.

[Update](#)

Retrieves and stylizes a raster.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: Preview Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Applies style traits to yield a preview of the specified raster.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMSEAnnotationStyle Class

[Classes](#)

Provides an annotation alteration style for stylization. Several properties can be represented as either a concrete type or as an ADE expression (in such cases, an API is provided for each representation). Setting a property by expression clears any value previously set by a concrete type. Likewise, setting any property by concrete type clears any value previously set in an expression. An API, `XxxIsExpression()` is provided to test which representation is currently in effect for each property. To represent the "" state, set the expression to NULL for that property. Using the wrong `GetXxx` function for the current representation of a property will return `Acad::eWrongObjectType`. The const `ACHAR *&` output parameters are valid immediately upon return and point directly to member data. The caller must not delete the pointer. If the caller desires to keep the value for an indefinite time, the caller must copy the output const `ACHAR*&` to its own memory and manage its lifetime.

```
class AcMapDMSEAnnotationStyle : public AcMapDMStylizationEntityStyle
File
```

DmSEAnnotationStyle.h

☐ Methods



[~AcMapDMSEAnnotationStyle](#) Destroys an instance of this class.



[AcMapDMSEAnnotationStyle](#) Constructs an instance of this class.

[clone](#)

Clones the object. See also `clone()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[CoExistenceType](#)

Styles may declare preferences for how they will co-exist with other styles under a single Element.

[ColorIsExpression](#)

Determines whether the color is an expression.

[copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[DeleteCookie](#)

Deletes the style's cookie during database destruction.

[Dismiss](#)

Removes topology stylization information that is used during regeneration.

[Dismiss](#)

Removes stylization information that is used during regeneration.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Enable](#)

Enables or disables the style for the specified topology.

[Enable](#)

Enables or disables the style for the specified entity.

[GetAnnotationTemplate](#)

Retrieves the annotation template ID.

<a href="#"><u>GetColor</u></a>	Retrieves the color.
<a href="#"><u>GetColorExpression</u></a>	Retrieves the color expression.
<a href="#"><u>GetLayer</u></a>	Retrieves the layer id.
<a href="#"><u>GetLayerExpression</u></a>	Retrieves the layer expression.
<a href="#"><u>GetLinetype</u></a>	Retrieves the linetype id.
<a href="#"><u>GetLinetypeExpression</u></a>	Retrieves the linetype expression.
<a href="#"><u>GetLineWeight</u></a>	Retrieves the lineweight.
<a href="#"><u>GetLineWeightExpression</u></a>	Retrieves the lineweight expression.
<a href="#"><u>GetLocationExpression</u></a>	Retrieves the location expression.
<a href="#"><u>GetRotation</u></a>	Retrieves the rotation.
<a href="#"><u>GetRotationExpression</u></a>	Retrieves the rotation expression.
<a href="#"><u>GetScale</u></a>	Retrieves the scale.
<a href="#"><u>GetScaleExpression</u></a>	Retrieves the scale expression.
<a href="#"><u>GetStylizationEntities</u></a>	Retrieves the AcDbObjectIds of any stylization entities created during an Update().
<a href="#"><u>IsSymbolStyle</u></a>	Indicates that this style is a symbol style instead of an annotation style.
<a href="#"><u>LayerIsExpression</u></a>	Determines whether the layer is an expression.
<a href="#"><u>LinetypeIsExpression</u></a>	Determines whether the linetype is an expression.
<a href="#"><u>LineWeightIsExpression</u></a>	Determines whether the lineweight is an expression.
<a href="#"><u>Preview</u></a>	Applies style traits to yield a preview of the specified entity.
<a href="#"><u>RotationIsExpression</u></a>	Determines whether the rotation is an expression.
<a href="#"><u>ScaleIsExpression</u></a>	Determines whether the scale is an expression.
<a href="#"><u>SetAnnotationTemplate</u></a>	Sets the annotation template ID or block ID.
<a href="#"><u>SetColor</u></a>	Sets the color.
<a href="#"><u>SetColorExpression</u></a>	Sets the color expression.
<a href="#"><u>SetLayer</u></a>	Sets the layer id.
<a href="#"><u>SetLayerExpression</u></a>	Sets the layer expression.

<a href="#">SetLinetype</a>	Sets the linetype id.
<a href="#">SetLinetypeExpression</a>	Sets the linetype expression.
<a href="#">SetLineWeight</a>	Sets the lineweight.
<a href="#">SetLineWeightExpression</a>	Sets the lineweight expression.
<a href="#">SetLocationExpression</a>	Sets the location expression.
<a href="#">SetRotation</a>	Sets the rotation.
<a href="#">SetRotationExpression</a>	Sets the rotation expression.
<a href="#">SetScale</a>	Sets the scale.
<a href="#">SetScaleExpression</a>	Sets the scale expression.
<a href="#">Update</a>	Builds topology stylization information for later regeneration.
<a href="#">Update</a>	Builds stylization information for later regeneration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: CoExistenceType Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Styles may declare preferences for how they will co-exist with other styles under a single Element.

```
virtual AcMapDMStyle::CoExistenceFlags CoExistenceType() const;
```

Returns

Returns Style's CoExistence preferences.

## Remarks

They may coexist with all other styles, only other styles of the same type, or may require being the only style under an Element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle::Dismiss Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Removes topology stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: Enable Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Enables or disables the style for the specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
pTopoName	Input topology name.
lTopoId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetColorExpression Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the color expression.

```
Acad::ErrorStatus GetColorExpression(  
    const ACHAR*& pColorExp  
) const;
```

Parameters	Description
pColorExp	Output color expression.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
AcMapDMSEAnnotationStyle:: GetLayerExpression Method  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the layer expression.

```
Acad::ErrorStatus GetLayerExpression(  
    const ACHAR*& pLayerExp  
) const;
```

Parameters	Description
pLayerExp	Output layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: GetLinetypeExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the linetype expression.

```
Acad::ErrorStatus GetLinetypeExpression(  
    const ACHAR*& pLinetypeExp  
) const;
```

Parameters	Description
pLinetypeExp	Output linetype expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: GetLineWeightExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the lineweight expression.

```
Acad::ErrorStatus GetLineWeightExpression(  
    const ACHAR*& pLineWeightExp  
) const;
```

Parameters	Description
pLineWeightExp	Output lineweight expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
AcMapDMSEAnnotationStyle:: GetLocationExpression Method  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the location expression.

```
Acad::ErrorStatus GetLocationExpression(  
    const ACHAR*& pLocationExp  
) const;
```

Parameters	Description
pLocationExp	Output location expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: GetRotationExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the rotation expression.

```
Acad::ErrorStatus GetRotationExpression(  
    const ACHAR*& pRotationExp  
) const;
```

Parameters	Description
pRotationExp	Output rotation expression.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: GetScaleExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the scale expression.

```
Acad::ErrorStatus GetScaleExpression(  
    const ACHAR*& pScaleExp  
) const;
```

Parameters	Description
pScaleExp	Output scale expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: Preview Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of IDs of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle::SetColorExpression Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the color expression.

```
Acad::ErrorStatus SetColorExpression(  
    const ACHAR* pColorExp  
);
```

Parameters	Description
pColorExp	Input color expression.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetLayerExpression Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the layer expression.

```
Acad::ErrorStatus SetLayerExpression(  
    const ACHAR* pLayerExp  
);
```

Parameters	Description
pLayerExp	Input layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: SetLinetypeExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the linetype expression.

```
Acad::ErrorStatus SetLinetypeExpression(  
    const ACHAR* pLinetypeExp  
);
```

Parameters	Description
pLinetypeExp	Input linetype expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: SetLineWeightExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the lineweight expression.

```
Acad::ErrorStatus SetLineWeightExpression(  
    const ACHAR* pLineWeightExp  
);
```

Parameters	Description
pLineWeightExp	Input lineweight expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: SetLocationExpression Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the location expression.

```
Acad::ErrorStatus SetLocationExpression(  
    const ACHAR* pLocationExp  
);
```

Parameters	Description
pLocationExp	Input location expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetRotationExpression Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the rotation expression.

```
Acad::ErrorStatus SetRotationExpression(  
    const ACHAR* pRotationExp  
);
```

Parameters	Description
pRotationExp	Input rotation expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: SetScaleExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the scale expression.

```
Acad::ErrorStatus SetScaleExpression(  
    const ACHAR* pScaleExp  
);
```

Parameters	Description
pScaleExp	Input scale expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: Update Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Builds topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMSEHatchStyle Class

[Classes](#)

Provides a hatch alteration style for stylization. Several properties can be represented as either a concrete type or as an ADE expression (in such cases, an API is provided for each representation). Setting a property by expression clears any value previously set by a concrete type. Likewise, setting any property by concrete type clears any value previously set in an expression. An API, `XxxIsExpression()` is provided to test which representation is currently in effect for each property. To represent the "" state, set the expression to NULL for that property. Using the wrong `GetXxx` function for the current representation of a property will return `Acad::eWrongObjectType`. The const ACHAR \*& output parameters are valid immediately upon return and point directly to member data. The caller must not delete the pointer. If the caller desires to keep the value for an indefinite time, the caller must copy the output const ACHAR\*& to its own memory and manage its lifetime.

```
class AcMapDMSEHatchStyle : public AcMapDMStylizationEntityStyle;  
File
```

DmSEHatchStyle.h

☐ Methods



[~AcMapDMSEHatchStyle](#) Destroys an instance of this class.



[AcMapDMSEHatchStyle](#) Constructs an instance of this class.

[clone](#) Clones the object. See also `clone()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[ColorIsExpression](#) Determines whether the color is an expression.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it

	directly.
<a href="#">DeleteCookie</a>	Deletes the style's cookie during database destruction.
<a href="#">Dismiss</a>	Removes topology stylization information that is used during regeneration.
<a href="#">Dismiss</a>	Removes stylization information that is used during regeneration.
<a href="#">dwgInFields</a>	Lets this object read its data. See also <code>dwgInFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dwgOutFields</a>	Lets this object write its data. See also <code>dwgOutFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dxfInFields</a>	Lets this object read its data. See also <code>dxfInFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">dxfOutFields</a>	Lets this object write its data. See also <code>dxfOutFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">Enable</a>	Enables or disables the style for the specified topology.
<a href="#">Enable</a>	Enables or disables the style for the specified entity.
<a href="#">GetColor</a>	Retrieves the color.
<a href="#">GetColorExpression</a>	Retrieves the color expression.
<a href="#">GetHatchPatternName</a>	Retrieves the hatch pattern name.
<a href="#">GetLayer</a>	Retrieves the layer id.
<a href="#">GetLayer</a>	Retrieves the layer name.

<a href="#"><u>GetLayerExpression</u></a>	Retrieves the layer expression.
<a href="#"><u>GetRotation</u></a>	Retrieves the rotation.
<a href="#"><u>GetRotationExpression</u></a>	Retrieves the rotation expression.
<a href="#"><u>GetScale</u></a>	Retrieves the scale.
<a href="#"><u>GetScaleExpression</u></a>	Retrieves the scale expression.
<a href="#"><u>GetStylizationEntities</u></a>	Retrieves the AcDbObjectIds of any stylization entities created during an Update().
<a href="#"><u>LayerIsExpression</u></a>	Determines whether the layer is an expression.
<a href="#"><u>Preview</u></a>	Applies style traits to yield a preview of the specified entity.
<a href="#"><u>RotationIsExpression</u></a>	Determines whether the rotation is an expression.
<a href="#"><u>ScaleIsExpression</u></a>	Determines whether the scale is an expression.
<a href="#"><u>SetColor</u></a>	Sets the color.
<a href="#"><u>SetColorExpression</u></a>	Sets the color expression.
<a href="#"><u>SetHatchPattern</u></a>	Sets the hatch pattern name.
<a href="#"><u>SetLayer</u></a>	Sets the layer name.
<a href="#"><u>SetLayer</u></a>	Sets the layer id.
<a href="#"><u>SetLayerExpression</u></a>	Sets the layer expression.
<a href="#"><u>SetRotation</u></a>	Sets the rotation.
<a href="#"><u>SetRotationExpression</u></a>	Sets the rotation expression.
<a href="#"><u>SetScale</u></a>	Sets the scale.
<a href="#"><u>SetScaleExpression</u></a>	Sets the scale expression.
<a href="#"><u>Update</u></a>	Builds topology stylization information for later regeneration.
<a href="#"><u>Update</u></a>	Builds stylization information for later regeneration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle::Dismiss Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Removes topology stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: Enable Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Enables or disables the style for the specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
pTopoName	Input topology name.
lTopoId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetColorExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the color expression.

```
Acad::ErrorStatus GetColorExpression(  
    const ACHAR*& pColorExp  
) const;
```

Parameters	Description
pColorExp	Output color expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetHatchPatternName Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the hatch pattern name.

```
Acad::ErrorStatus GetHatchPatternName(  
    const ACHAR*& pHatchPatternName  
) const;
```

Parameters	Description
pHatchPatternName	Output hatch pattern name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetLayer Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the layer name.

```
Acad::ErrorStatus GetLayer(  
    const ACHAR*& pLayerName  
) const;
```

Parameters	Description
pLayerName	Output layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetLayerExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the layer expression.

```
Acad::ErrorStatus GetLayerExpression(  
    const ACHAR*& pLayerExp  
) const;
```

Parameters	Description
pLayerExp	Output layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetRotationExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the rotation expression.

```
Acad::ErrorStatus GetRotationExpression(  
    const ACHAR*& pRotationExp  
) const;
```

Parameters	Description
pRotationExp	Output rotation expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetScaleExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the scale expression.

```
Acad::ErrorStatus GetScaleExpression(  
    const ACHAR*& pScaleExp  
) const;
```

Parameters	Description
pScaleExp	Output scale expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: Preview Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of IDs of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetColorExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the color expression.

```
Acad::ErrorStatus SetColorExpression(  
    const ACHAR* pColorExp  
);
```

Parameters	Description
pColorExp	Input color expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetHatchPattern Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the hatch pattern name.

```
Acad::ErrorStatus SetHatchPattern(  
    const ACHAR* pHatchPatternName  
);
```

Parameters	Description
pHatchPatternName	Input hatch pattern name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetLayer Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the layer name.

```
Acad::ErrorStatus SetLayer(  
    const ACHAR* pLayerName  
);
```

Parameters	Description
pLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetLayerExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the layer expression.

```
Acad::ErrorStatus SetLayerExpression(  
    const ACHAR* pLayerExp  
);
```

Parameters	Description
pLayerExp	Input layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetRotationExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the rotation expression.

```
Acad::ErrorStatus SetRotationExpression(  
    const ACHAR* pRotationExp  
);
```

Parameters	Description
pRotationExp	Input rotation expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetScaleExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the scale expression.

```
Acad::ErrorStatus SetScaleExpression(  
    const ACHAR* pScaleExp  
);
```

Parameters	Description
pScaleExp	Input scale expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: Update Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Builds topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMSETextStyle Class

[Classes](#)

Provides a text alteration style for stylization. Several properties can be represented as either a concrete type or as an ADE expression (in such cases, an API is provided for each representation). Setting a property by expression clears any value previously set by a concrete type. Likewise, setting any property by concrete type clears any value previously set in an expression. An API, `XxxIsExpression()` is provided to test which representation is currently in effect for each property. To represent the "" state, set the expression to NULL for that property. Using the wrong `GetXxx` function for the current representation of a property will return `Acad::eWrongObjectType`. The const ACHAR \*& output parameters are valid immediately upon return and point directly to member data. The caller must not delete the pointer. If the caller desires to keep the value for an indefinite time, the caller must copy the output const ACHAR\*& to its own memory and manage its lifetime.

```
class AcMapDMSETextStyle : public AcMapDMStylizationEntityTypeStyle;  
File
```

DmSETextStyle.h

☐ Enumerations

📄 [Justification](#)

Enumerates the types of text justification.

☐ Methods



[~AcMapDMSETextStyle](#)

Destroys an instance of this class.



[AcMapDMSETextStyle](#)

Constructs an instance of this class.

[clone](#)

Clones the object. See also `clone()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[ColorIsExpression](#)

Determines whether the color is an expression.

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the

[copyFrom](#)

AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[DeleteCookie](#)

Deletes the style's cookie during database destruction.

[Dismiss](#)

Removes topology stylization information that is used during regeneration.

[Dismiss](#)

Removes stylization information that is used during regeneration.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Enable](#)

Enables or disables the style for the specified topology.

[Enable](#)

Enables or disables the style for the specified entity.

[GetColor](#)

Retrieves the text color.

[GetColorExpression](#)

Retrieves the color expression.

[GetHeight](#)

Retrieves the text height.

<a href="#"><u>GetHeightExpression</u></a>	Retrieves the height expression.
<a href="#"><u>GetJustification</u></a>	Retrieves the text justification.
<a href="#"><u>GetJustificationExpression</u></a>	Retrieves the justification expression.
<a href="#"><u>GetLayer</u></a>	Retrieves the text layer id.
<a href="#"><u>GetLayer</u></a>	Retrieves the text layer.
<a href="#"><u>GetLayerExpression</u></a>	Retrieves the layer expression.
<a href="#"><u>GetLocationExpression</u></a>	Retrieves the location expression.
<a href="#"><u>GetRotation</u></a>	Retrieves the text rotation.
<a href="#"><u>GetRotationExpression</u></a>	Retrieves the rotation expression.
<a href="#"><u>GetStyle</u></a>	Retrieves the text style id.
<a href="#"><u>GetStyle</u></a>	Retrieves the text style name.
<a href="#"><u>GetStyleExpression</u></a>	Retrieves the style expression.
<a href="#"><u>GetStylizationEntities</u></a>	Retrieves the AcDbObjectIds of any stylization entities created during an Update().
<a href="#"><u>GetTextStringExpression</u></a>	Retrieves the text string expression.
<a href="#"><u>HeightIsExpression</u></a>	Determines whether the height is an expression.
<a href="#"><u>JustificationIsExpression</u></a>	Determines whether justification is an expression.
<a href="#"><u>LayerIsExpression</u></a>	Determines whether the layer is an expression.
<a href="#"><u>Preview</u></a>	Applies style traits to yield a preview of the specified entity.
<a href="#"><u>RotationIsExpression</u></a>	Determines whether the rotation is an expression.
<a href="#"><u>SetColor</u></a>	Sets the text color.
<a href="#"><u>SetColorExpression</u></a>	Sets the color expression.
<a href="#"><u>SetHeight</u></a>	Sets the text height.
<a href="#"><u>SetHeightExpression</u></a>	Sets the height expression.
<a href="#"><u>SetJustification</u></a>	Sets the text justification.
<a href="#"><u>SetJustificationExpression</u></a>	Sets the justification expression.
<a href="#"><u>SetLayer</u></a>	Sets the layer by name.
<a href="#"><u>SetLayer</u></a>	Sets the layer by id.
<a href="#"><u>SetLayerExpression</u></a>	Sets the layer expression.
<a href="#"><u>SetLocationExpression</u></a>	Sets the location expression.
<a href="#"><u>SetRotation</u></a>	Sets the text rotation.
<a href="#"><u>SetRotationExpression</u></a>	Sets the rotation expression.

<a href="#">SetStyle</a>	Sets the text style by name.
<a href="#">SetStyle</a>	Sets the text style by id.
<a href="#">SetStyleExpression</a>	Sets the style expression.
<a href="#">SetTextStringExpression</a>	Sets the text string expression.
<a href="#">StyleIsExpression</a>	Determines whether the style is an expression.
<a href="#">Update</a>	Builds topology stylization information for later regeneration.
<a href="#">Update</a>	Builds stylization information for later regeneration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle::Dismiss Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Removes topology stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: Enable Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Enables or disables the style for the specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
pTopoName	Input topology name.
lTopoId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: GetColorExpression Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the color expression.

```
Acad::ErrorStatus GetColorExpression(  
    const ACHAR*& pColorExp  
) const;
```

Parameters	Description
pColorExp	Output color expression.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetHeightExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the height expression.

```
Acad::ErrorStatus GetHeightExpression(  
    const ACHAR*& pHeightExp  
) const;
```

Parameters	Description
pHeightExp	Output height expression.

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: GetJustificationExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the justification expression.

```
Acad::ErrorStatus GetJustificationExpression(  
    const ACHAR*& pJustificationExp  
) const;
```

Parameters	Description
pJustificationExp	Output justification expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetLayer Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text layer.

```
Acad::ErrorStatus GetLayer(  
    const ACHAR*& pLayerName  
) const;
```

Parameters	Description
pLayerName	Output layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: GetLayerExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the layer expression.

```
Acad::ErrorStatus GetLayerExpression(  
    const ACHAR*& pLayerExp  
) const;
```

Parameters	Description
pLayerExp	Output layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: GetLocationExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the location expression.

```
Acad::ErrorStatus GetLocationExpression(  
    const ACHAR*& pLocationExpression  
) const;
```

Parameters	Description
pLocationExpression	Output location expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: GetRotationExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the rotation expression.

```
Acad::ErrorStatus GetRotationExpression(  
    const ACHAR*& pRotationExp  
) const;
```

Parameters	Description
pRotationExp	Output rotation expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetStyle Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text style name.

```
Acad::ErrorStatus GetStyle(  
    const ACHAR*& pStyleName  
) const;
```

Parameters	Description
pStyleName	Output style name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: GetStyleExpression Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the style expression.

```
Acad::ErrorStatus GetStyleExpression(  
    const ACHAR*& pStyleExp  
) const;
```

Parameters	Description
pStyleExp	Output style expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: GetTextStringExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text string expression.

```
Acad::ErrorStatus GetTextStringExpression(  
    const ACHAR*& pTextExpression  
) const;
```

Parameters	Description
pTextExpression	Output text string expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: Preview Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of IDs of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetColorExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the color expression.

```
Acad::ErrorStatus SetColorExpression(  
    const ACHAR* pColorExp  
);
```

Parameters	Description
pColorExp	Input color expression.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetHeightExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the height expression.

```
Acad::ErrorStatus SetHeightExpression(  
    const ACHAR* pHeightExp  
);
```

Parameters	Description
pHeightExp	Input height string expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: SetJustificationExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the justification expression.

```
Acad::ErrorStatus SetJustificationExpression(  
    const ACHAR* pJustificationExp  
);
```

Parameters	Description
pJustificationExp	Input justification expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetLayer Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the layer by name.

```
Acad::ErrorStatus SetLayer(  
    const ACHAR* pLayerName  
);
```

Parameters	Description
pLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetLayerExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the layer expression.

```
Acad::ErrorStatus SetLayerExpression(  
    const ACHAR* pLayerExp  
);
```

Parameters	Description
pLayerExp	Input layer expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: SetLocationExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the location expression.

```
Acad::ErrorStatus SetLocationExpression(  
    const ACHAR* pLocationExp  
);
```

Parameters	Description
pLocationExp	Input location expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
[AcMapDMSETextStyle:: SetRotationExpression Method](#)  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the rotation expression.

```
Acad::ErrorStatus SetRotationExpression(  
    const ACHAR* pRotationExp  
);
```

Parameters	Description
pRotationExp	Input rotation expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetStyle Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text style by name.

```
Acad::ErrorStatus SetStyle(  
    const ACHAR* pStyleName  
);
```

Parameters	Description
pStyleName	Input style name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetStyleExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the style expression.

```
Acad::ErrorStatus SetStyleExpression(  
    const ACHAR* pStyleExp  
);
```

Parameters	Description
pStyleExp	Input style expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: SetTextStringExpression Method  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text string expression.

```
Acad::ErrorStatus SetTextStringExpression(  
    const ACHAR* pTextExp  
);
```

Parameters	Description
pTextExp	Input string expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: Update Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Builds topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
pTopoName	Input topology name.
lTopoId	Input topology element ID.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMStyle Class

[Classes](#)

Base class for display-management styles.

```
class AcMapDMStyle : public AcDbObject;  
File
```

DmDisplayStyle.h

Remarks

This class includes utility functions that stylize, unstylize, cache style values, preview, and so on.

☐ Enumerations

 [CachableProperties](#) Enumerates ADE non-geometric DOT variables.

 [CoExistenceFlags](#) Enumerates CoExistence preferences that Styles may declare.

☐ Methods

[Apply](#) Applies this style to a specified entity.

[CacheStylizedPropValue](#) Caches the stylized string property value.

[CacheStylizedPropValue](#) Caches the stylized color property value.

[CacheStylizedPropValue](#) Caches the stylized integer property value.

[CacheStylizedPropValue](#) Caches the stylized topology string property value.

[CacheStylizedPropValue](#) Caches the stylized topology color property value.

[CacheStylizedPropValue](#) Caches the stylized topology integer property value.

[CoExistenceType](#) Styles may declare preferences for how they will co-exist with other styles under a single Element.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Decrements the reference count to indicate there is

## [DecrementRef](#)

one fewer client using this style. If the reference count becomes zero, the style erases itself. You do not need to call this function directly unless you are implementing an [AcMapDMStyleReference](#) class. See also [IncrementRef\(\)](#).

## [deepClone](#)

Clones this style. See also [deepClone\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

## [DeleteCookie](#)

Deletes the style's cookie during database destruction.

## [Dismiss](#)

Removes the stylization information that is used during regeneration.

## [Dismiss](#)

Removes the stylization information that is used during regeneration.

## [DisplayName](#)

Gets an alternative, constant name for this type of style in certain situations where the instance name may be unavailable.

## [dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

## [dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

## [dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

## [dxfOutFields](#)

Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

## [Enable](#)

Enables or disables the style for the specified

	topology.
<a href="#">Enable</a>	Enables or disables the style for the specified entity.
<a href="#">GetName</a>	Retrieves the name of this style. See also <a href="#">SetName()</a> .
<a href="#">GetStylizationEntities</a>	Retrieves the <a href="#">AcDbObjectIds</a> of any stylization entities created during an <a href="#">Update()</a> .
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">IncrementRef</a>	Increments the reference count to indicate that there is one more client using this style. You do not need to call this function directly unless you are implementing an <a href="#">AcMapDMStyleReference</a> class. The caller must call <a href="#">DecrementRef()</a> when finished with this style.
<a href="#">IsMultiplyReferenced</a>	Determines whether this style has multiple references.
<a href="#">OnMapProjectInitialized</a>	Invoked when the an AutoCAD Map project is initialized.
<a href="#">OnObjectAppended</a>	Invoked after a style has been appended to the database, allowing the appended style to do any necessary post-append initialization.
<a href="#">Preview</a>	Stylizes an entity.
<a href="#">Project</a>	Retrieves the AutoCAD Map project.
<a href="#">Refresh</a>	Refreshes stylization information for later regeneration.
<a href="#">Refresh</a>	Refreshes topology stylization information for later regeneration.
<a href="#">SetName</a>	Sets the name of this style. See also <a href="#">GetName()</a> .
<a href="#">subClose</a>	Invoked from within <a href="#">close()</a> before the close actually occurs. The default implementation of this function returns <a href="#">Acad::eOk</a> . See also <a href="#">subClose()</a> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">subErase</a>	Invoked from within <a href="#">erase()</a> before the erase actually occurs. See also <a href="#">subErase()</a> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[UnApply](#)

Removes the style from a specified entity.

[Update](#)

Builds topology stylization information for later regeneration.

[Update](#)

Builds stylization information for later regeneration.

[wblockClone](#)

Clones this style. See also `wblockClone()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#)

AcMapDMStyle:: CoExistenceFlags Enumeration

[AcMapDMStyle Class](#)

Enumerates CoExistence preferences that Styles may declare.

```
enum CoExistenceFlags {  
    kNotApplicable = 0,  
    kAllStyles,  
    kSingleton,  
    kLikeKind  
};
```

File

DmDisplayStyle.h

Parameters	Description
kNotApplicable	A default/uninitialized value.
kAllStyles	This style will exist under an Element with all other style types.
kSingleton	This style must be the only style under an Element.
kLikeKind	This style can only exist under an Element with other styles of the same kind.

Remarks

This state is considered by the Element during AddStyle.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: CacheStylizedPropValue Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized string property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    AcDbObjectId entityId,  
    CachableProperties key,  
    const ACHAR* value  
) const;
```

Parameters	Description
entityId	Input ID of the entity.
key	Input CachablePropertieskey.
value	Input string value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: CacheStylizedPropValue Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized topology string property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    CachableProperties key,  
    const ACHAR* value  
) const;
```

Parameters	Description
pTopoName	Input topology name.
lTopoId	Input topology ID.
key	Input CachablePropertieskey.
value	Input string value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: CacheStylizedPropValue Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized topology color property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    CachableProperties key,  
    const AcCmColor& color  
) const;
```

Parameters	Description
pTopoName	Input topology name.
lTopoId	Input topology ID.
key	Input CachablePropertieskey.
color	Input color value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: CacheStylizedPropValue Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized topology integer property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    CachableProperties key,  
    int value  
) const;
```

Parameters	Description
pTopoName	Input topology name.
lTopoId	Input topology ID.
key	Input CachablePropertieskey.
value	Input integer value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)  
AcMapDMStyle:: CoExistenceType Method  
[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Styles may declare preferences for how they will co-exist with other styles under a single Element.

**virtual** [AcMapDMStyle::CoExistenceFlags](#) CoExistenceType() **const**;  
Returns

Returns Style's CoExistence preferences.

## Remarks

They may coexist with all other styles, only other styles of the same type, or may require being the only style under an Element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: deepClone Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Clones this style. See also deepClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus deepClone(  
    AcDbObject* pOwnerObject,  
    AcDbObject*& pClonedObject,  
    AcDbIdMapping& idMap,  
    Adesk::Boolean isPrimary = true  
) const;
```

Parameters	Description
pOwnerObject	Input object to append the clone to.
pClonedObject	Returned cloned object, or NULL if not cloned.
idMap	Input current object map ID.
isPrimary	Input true if this object is primary, or false if it is owned.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Dismiss Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Removes the stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: DisplayName Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Gets an alternative, constant name for this type of style in certain situations where the instance name may be unavailable.

```
virtual const ACHAR* DisplayName() const;
```

Returns

Returns a const ACHAR\* alternative name for certain display purposes.

Remarks

Derived classes should return a constant, generally descriptive name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Enable Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Enables or disables the style for the specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: Preview Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Stylizes an entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse = 0  
);
```

Parameters	Description
createdEntities	Output array of IDs of any created entities.
targetEntId	Input ID of the entity to be stylized.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If any additional entities are created as a result of stylization, as in the case of text, hatch, and annotation alterations, then createdEntities will contain them. The subsequent lifetime of the created entities is managed by the preview system. There is no need to do any caching or other base-class messaging for stylization of preview entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: Refresh Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Refreshes stylization information for later regeneration.

```
virtual Acad::ErrorStatus Refresh(  
    void*& pCookie,  
    AcDbObjectId entityId,  
    Adesk::UInt32 flag = 0  
);
```

### Parameters

### Description

pCookie

Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.

entityId

Input object ID.

flag

Reserved. For internal use only.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Remarks

Derived classes can reevaluate any Update()-time actions here. An example would be to rebuild a stylization entity in response to changes in the target entity. If the Style populated the cookie during Update, the appropriate cookie will be provided for the target entity. It is acceptable for this method to be a no-op.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Refresh Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Refreshes topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Refresh(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can reevaluate any Update()-time actions here. An example would be to rebuild a stylization entity in response to changes in the target entity. If the Style populated the cookie during Update, the appropriate cookie will be provided for the target topology. It is acceptable for this method to be a no-op.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle::SetName Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Sets the name of this style. See also GetName().

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR* pName  
);
```

Parameters	Description
pName	Input style name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Update Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Builds topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMStyleCategory Class

[Classes](#)

Represents a category of styles.

```
class AcMapDMStyleCategory : public AcDbObject;  
File
```

DmStyleCategory.h

☐ Methods



[~AcMapDMStyleCategory](#) Destroys an instance of this class.



[AcMapDMStyleCategory](#) Constructs an instance of this class.

[Append](#) Adds a style to this category.

[AppendStyleCopy](#) Copies a style and appends it to this category.

[AppendStyleCopy](#) Copies multiple styles and appends them to this category.

[audit](#) This function is called by AutoCAD when the AUDIT command is executed.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[erased](#)

Lets this object listen to erase-notifications from items that it owns. See also `erased()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Find](#)

Finds a specified style within this category.

[GetAt](#)

Retrieves the ID of a style at a specified position.

[GetAt](#)

Retrieves a style at a specified position, opened in a specified mode.

[GetAt](#)

Retrieves a style, opened in a specified mode.

[GetName](#)

Retrieves the name of this category. See also `SetName()`.

[Has](#)

Determines whether this category has a specified style.

[Implementation](#)

Returns the implementation object.

[InsertAt](#)

Inserts a style at a specified position.

[InsertStyleCopyAt](#)

Inserts a copy of a style at a specified position.

[InsertStyleCopyAt](#)

Inserts multiple styles at a specified position.

[Move](#)

Moves a style to a new position.

[Move](#)

Moves a style to a new position.

[NumStyles](#)

Counts the number of styles within this category.

[Remove](#)

Removes a style.

[RemoveAt](#)

Removes a style at a specified position.

[SetName](#)

Sets the name of this category. See also `GetName()`.

[subClose](#)

Invoked from within `close()` before the close actually occurs. The default implementation of this function returns `Acad::eOk`. See also `subClose()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as

needed; it is unlikely that you will need to call it directly.

### [subErase](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory::SetName Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Sets the name of this category. See also GetName().

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input category name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMStyleLibrary Class

[Classes](#)

Represents a library of categories of styles.

```
class AcMapDMStyleLibrary : public AcDbObject;
```

File

DmStyleLibrary.h

☐ Methods



[~AcMapDMStyleLibrary](#) Destroys an instance of this class.



[AcMapDMStyleLibrary](#) Constructs an instance of this class.

[Append](#) Adds a style category to this library.

[audit](#) This function is called by AutoCAD when the AUDIT command is executed.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#) Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is

called by the system as needed; it is unlikely that you will need to call it directly.

<a href="#">erased</a>	Lets this object listen to erase-notifications from items that it owns. See also <code>erased()</code> in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#">GetAt</a>	Retrieves a category at a specified position.
<a href="#">GetAt</a>	Retrieves a category at a specified position, opened in a specified mode.
<a href="#">GetAt</a>	Retrieves a category, opened in a specified mode.
<a href="#">Has</a>	Determines whether a specific category exists in this library.
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">InsertAt</a>	Inserts a category at a specified position.
<a href="#">Move</a>	Moves a category from one position to another in this library.
<a href="#">Move</a>	Moves a category with a specified ID to another position in this library.
<a href="#">NumCategories</a>	Counts the number of categories within this library.
<a href="#">Remove</a>	Removes a category from this library.
<a href="#">RemoveAt</a>	Removes a category at a specified position from this library.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMStyleReference Class

[Classes](#)

Represents a reference to an existing style.

```
class AcMapDMStyleReference : public AcDbObject;
```

File

DmStyleReference.h

Remarks

These references create additional entities in the database, rather than modifying the appearance of existing entities.

☐ Methods



[~AcMapDMStyleReference](#) Destroys an instance of this class.



[AcMapDMStyleReference](#) Constructs an instance of this class.

[audit](#)

This function is called by AutoCAD when the AUDIT command is executed.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutfields](#)

Lets this object write its data. See also `dxfoutfields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Implementation](#)

Returns the implementation object.

[IsEnabled](#)

Determines whether this reference is enabled. See also `setEnabled()`.

[IsUniqueReference](#)

Determines whether this reference is the only reference to the style that this reference refers to. See also `makeUniqueReference()`.

[MakeUniqueReference](#)

Determines whether this reference is a unique reference and, if not, clones the referenced style and resets the style ID to the new copy. See also `isUniqueReference()`.

[setEnabled](#)

Enables or disables this reference. See also `isEnabled()`.

[StyleId](#)

Retrieves the ID of the style that this reference refers to.

[subClose](#)

Invoked from within `close()` before the close actually occurs. The default implementation of this function returns `Acad::eOk`. See also `subClose()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[subErase](#)

Invoked from within `erase()` before the erase actually occurs. See also `subErase()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[wblockClone](#)

Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object `pOwnerObject`. See also `wblockClone()` in the AutoCAD ObjectARX Developer's Guide. This

overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMStylizationEntityType Class

[Classes](#)

Represents a style of a stylization entity.

```
class AcMapDMStylizationEntityType : public AcMapDMStyle;  
File
```

DmStyleEntStyle.h

Remarks

A stylization entity is a new entity that stylization generates and adds to the database (rather than making temporary modifications during regeneration). This abstract class is not intended for direct instantiation.

☐ Methods

[Apply](#)

Applies the style to an entity.

[CacheStylizationEntity](#)

Informs the display manager that a particular AcDbEntity was created as a result of applying a StylizationEntityType.

[Dismiss](#)

Removes topology stylization information that is used during regeneration.

[Dismiss](#)

Removes stylization information that is used during regeneration.

[dwgInFields](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See

[dwgOutFields](#)

also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

[Enable](#)

Enables or disables the style for the specified topology.

[Enable](#)

Enables or disables the style for the specified entity.

[ExcludeStylizationEntityFromHatchStyle](#)

Creates an island around the stylization entity.

[ExcludeStylizationEntityFromHatchStyle](#)

Creates an island around the stylization entity.

[GetStylizationEntities](#)

Retrieves the `AcDbObjectIds` of any stylization entities created during an `Update()`.

[Preview](#)

Applies style traits to yield a preview of the specified entity. Builds stylization information for

[UnApply](#)

later regeneration.

[UnExcludeStylizationEntityFromHatchStyle](#)

A derived class must call this function to remove a stylization entity from association with a target topology, for the purposes of AcMapDMSEHatchStyle islanding.

[UnExcludeStylizationEntityFromHatchStyle](#)

A derived class must call this function to remove a stylization entity from association with a target entity, for the purposes of AcMapDMSEHatchStyle islanding.

[Update](#)

Builds topology stylization information for later regeneration  
Derived classes can perform any one-time Update()-time actions here.

[Update](#)

Builds stylization information for later regeneration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)

[AcMapDMStylizationEntityStyle::Dismiss Method](#)

[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Removes topology stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: Enable Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Enables or disables the style for the specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle::](#)  
[ExcludeStylizationEntityFromHatchStyle Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Creates an island around the stylization entity.

```
Acad::ErrorStatus ExcludeStylizationEntityFromHatchStyle(  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    const AcDbObjectId styleEntId  
) const;
```

Parameters	Description
pTopoName	Input topology name.
lTopoId	Input topology ID.
styleEntId	Input ID of the stylization entity.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

A derived class must call this function if it creates a stylization entity, and wants the AcMapDMSEHatchStyle to attempt to create an island around the stylization entity, should a stylization hatch happen to be applied to the same target topology. In this release of AutoCAD Map, islanding works for only AcDbText entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: Preview Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Applies style traits to yield a preview of the specified entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray& createdEntities,  
    AcDbObjectId targetEntId,  
    void* internalUse  
);
```

Parameters	Description
createdEntities	Output array of IDs of created alteration entities.
targetEntId	Input ID of the preview entity.
internalUse	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

For stylization entity styles such as text, hatch, and annotation, this function returns an array of created entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)

[AcMapDMStylizationEntityStyle::](#)

[UnExcludeStylizationEntityFromHatchStyle Method](#)

[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

A derived class must call this function to remove a stylization entity from association with a target topology, for the purposes of AcMapDMSEHatchStyle islanding.

```
Acad::ErrorStatus UnExcludeStylizationEntityFromHatchStyle(  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoId,  
    const AcDbObjectId styleEntId  
) const;
```

Parameters	Description
pTopoName	Input topology name.
lTopoId	Input topology ID.
styleEntId	Input ID of the stylization entity.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This should be done by the Style at dismiss time if the Style may reuse the same stylization entity against a different target topology. In this release of AutoCAD Map, islanding works for only AcDbText entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: Update Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Builds topology stylization information for later regeneration Derived classes can perform any one-time Update()-time actions here.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

### Parameters

### Description

pCookie

Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.

pTopoName

Input topology name.

lTopoElemId

Input topology element ID.

flag

Reserved. For internal use only.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Remarks

At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicStyle Class

[Classes](#)

Represents a thematic stylization.

```
class AcMapDMThematicStyle : public AcMapDMStyle;
```

File

DmThematicStyle.h

☐ Methods

⇒ [~AcMapDMThematicStyle](#)

Destroys an instance of this class.

⇒ [AcMapDMThematicStyle](#)

Constructs an instance of this class.

[Apply](#)

Applies the style to an entity.

[AutoRecalculateRanges](#)

Determines whether automatic recalculation of ranges is turned on or off.

[BuildRangeTables](#)

Builds range tables.

[CoExistenceType](#)

Styles may declare preferences for how they will co-exist with other styles under a single Element.

[DataSourceExpression](#)

Retrieves the expression used in theming.

[DeleteCookie](#)

Deletes the style's cookie during database destruction.

[Dismiss](#)

Removes topology stylization information that is used during regeneration.

[Dismiss](#)

Removes stylization information that is used during regeneration.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also

[dwgOutFields](#)

dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object read its data. See also dxfinFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also dxfoutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[Enable](#)

Enables or disables the style for a specified topology.

[Enable](#)

Enables or disables the style for a specified entity.

[GetStylizationEntities](#)

Retrieves the IDs of any stylization entities created during an Update().

[GetThematicTable](#)

Retrieves the thematic table, opened in a specified mode.

[GroupCount](#)

Counts the number of groups for thematic ranges.

[GroupingAlgorithmName](#)

Retrieves the name of the grouping algorithm.

[GroupRoundingMethod](#)

Retrieves the name of the rounding method used.

[IsAlterAnnotationEnabled](#)

Determines whether annotation alteration is set to modify a thematic stylization query result.

[IsAlterBlockEnabled](#)

Determines whether block alteration is set to modify a thematic stylization query result.

[IsAlterColorEnabled](#)

Determines whether color alteration is set to modify a thematic stylization query result.

[IsAlterHatchEnabled](#)

Determines whether hatch alteration is set to modify a thematic stylization query result.

[IsAlterLinestyleEnabled](#)

Determines whether linestyle alteration is set to modify a thematic stylization query result.

[IsAlterLinetypeEnabled](#)

Determines whether linetype alteration is set to modify a thematic stylization query result.

[IsAlterLineweightEnabled](#)

Determines whether lineweight alteration is set to modify a thematic stylization query result.

[IsAlterPlotstyleEnabled](#)

Determines whether plotstyle alteration is set to modify a thematic stylization query result.

[IsAlterTextEnabled](#)

Determines whether text alteration is set to modify a thematic stylization query result.

[IsScaleColorRamp](#)

Determines whether the scale-color-ramp flag is set.

[IsScaleHatchRamp](#)

Determines whether the scale-hatch-ramp flag is set.

[IsScaleLinestyleRamp](#)

Determines whether the scale-linestyle-ramp flag is set.

[IsScaleRampToFit](#)

Determines whether the scale-ramp-to-fit flag is set.

[IsSourceDataNumeric](#)

Determines whether the data source is numeric.

[LastColorRampName](#)

Retrieves the name of the last-used color ramp.

[LastHatchRampName](#)

Retrieves the name of the last-used hatch ramp.

[LastLinestyleRampName](#)

Retrieves the name of the last-used

<a href="#"><u>LinetypeScale</u></a>	linestyle ramp. Retrieves the linetype scale.
<a href="#"><u>NormalizationExpression</u></a>	Retrieves the normalization expression.
<a href="#"><u>OnObjectAppended</u></a>	Invoked when an object is assigned to the database.
<a href="#"><u>Preview</u></a>	Stylizes an entity.
<a href="#"><u>RangeTableComparisonOperator</u></a>	Retrieves the range-table comparison operator.
<a href="#"><u>SetAlterAnnotation</u></a>	Sets annotation alteration.
<a href="#"><u>SetAlterBlock</u></a>	Sets block alteration.
<a href="#"><u>SetAlterColor</u></a>	Sets color alteration.
<a href="#"><u>SetAlterHatch</u></a>	Sets hatch alteration.
<a href="#"><u>SetAlterLinestyle</u></a>	Sets linestyle alteration.
<a href="#"><u>SetAlterLinetype</u></a>	Sets linetype alteration.
<a href="#"><u>SetAlterLineweight</u></a>	Sets lineweight alteration.
<a href="#"><u>SetAlterPlotstyle</u></a>	Sets plotstyle alteration.
<a href="#"><u>SetAlterText</u></a>	Sets text alteration.
<a href="#"><u>SetAutoRecalculateRanges</u></a>	Turns automatic recalculation on or off.
<a href="#"><u>SetDataSourceExpression</u></a>	Sets the expression to use in theming.
<a href="#"><u>SetGroupCount</u></a>	Sets the number of groups for thematic ranges.
<a href="#"><u>SetGroupingAlgorithmName</u></a>	Sets the grouping algorithm.
<a href="#"><u>SetGroupRoundingMethod</u></a>	Sets the rounding method to use.
<a href="#"><u>SetLastColorRampName</u></a>	Sets the last-used color ramp.
<a href="#"><u>SetLastHatchRampName</u></a>	Sets the last-used hatch ramp.
<a href="#"><u>SetLastLinestyleRampName</u></a>	Sets the last-used linestyle ramp.
<a href="#"><u>SetLinetypeScale</u></a>	Sets the linetype scale.
<a href="#"><u>SetNormalizationExpression</u></a>	Sets the normalization expression.
<a href="#"><u>SetRangeTableComparisonOperator</u></a>	Sets the range-table comparison operator.
<a href="#"><u>SetScaleColorRamp</u></a>	Sets the scale-color-ramp flag.
<a href="#"><u>SetScaleHatchRamp</u></a>	Sets the scale-hatch-ramp flag.
<a href="#"><u>SetScaleLinestyleRamp</u></a>	Sets the scale-linestyle-ramp flag.
<a href="#"><u>SetScaleRampToFit</u></a>	Sets the scale-ramp-to-fit flag.

[SetSourceDataNumeric](#)

Sets the data source of the thematic stylization to be numeric.

[SetUseThousandsSeparators](#)

Sets the use of thousands separators.

[SetValuesToIgnore](#)

Sets the values-to-ignore expression.

[subErase](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[UnApply](#)

Builds stylization information for later regeneration.

[Update](#)

Builds topology stylization information for later regeneration.

[Update](#)

Builds stylization information for later regeneration.

[UseThousandsSeparators](#)

Determines whether thousands separators are used.

[ValuesToIgnore](#)

Retrieves the values-to-ignore expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: CoExistenceType Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Styles may declare preferences for how they will co-exist with other styles under a single Element.

```
virtual AcMapDMStyle::CoExistenceFlags CoExistenceType() const;
```

Returns

Returns Style's CoExistence preferences.

Remarks

The ThematicStyle must be the only style under a given Element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle::Dismiss Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Removes topology stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Enable Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Enables or disables the style for a specified topology.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Preview Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Stylizes an entity.

```
virtual Acad::ErrorStatus Preview(  
    AcDbObjectIdArray & createdEntities,  
    AcDbObjectId targetEntity,  
    void* rangeIndex  
);
```

Parameters	Description
createdEntities	Output array of IDs of any created entities.
targetEntity	Input ID of the entity to be stylized.
rangeIndex	Input range index.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If any additional entities are created as a result of stylization, as in the case of text, hatch, and annotation alterations, then createdEntities will contain them. The subsequent lifetime of the created entities is managed by the preview system. There is no need to do any caching or other base-class messaging for stylization of preview entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetDataSourceExpression Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the expression to use in theming.

```
virtual Acad::ErrorStatus SetDataSourceExpression(  
    const ACHAR* pszSourceExpression  
);
```

Parameters	Description
pszSourceExpression	Input expression string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
[AcMapDMThematicStyle:: SetGroupingAlgorithmName Method](#)  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the grouping algorithm.

```
virtual Acad::ErrorStatus SetGroupingAlgorithmName(  
    const ACHAR* pszGroupingAlgorithmName  
);
```

Parameters

Description

pszGroupingAlgorithmName Input name of the grouping algorithm.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
AcMapDMThematicStyle:: SetGroupRoundingMethod Method  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the rounding method to use.

```
virtual Acad::ErrorStatus SetGroupRoundingMethod(  
    const ACHAR* pszGroupRoundingMethod  
);
```

Parameters

Description

pszGroupRoundingMethod Input name of the rounding method.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetLastColorRampName Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the last-used color ramp.

```
virtual Acad::ErrorStatus SetLastColorRampName(  
    const ACHAR* pszLastColorRampName  
);
```

Parameters	Description
pszLastColorRampName	Input ramp name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetLastHatchRampName Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the last-used hatch ramp.

```
virtual Acad::ErrorStatus SetLastHatchRampName(  
    const ACHAR* pszLastHatchRampName  
);
```

Parameters	Description
pszLastHatchRampName	Input ramp name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
[AcMapDMThematicStyle:: SetLastLineStyleRampName Method](#)  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the last-used linestyle ramp.

```
virtual Acad::ErrorStatus SetLastLineStyleRampName(  
    const ACHAR* pszLastLineStyleRampName  
);
```

Parameters	Description
pszLastLineStyleRampName	Input ramp name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
AcMapDMThematicStyle:: SetNormalizationExpression Method  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the normalization expression.

```
virtual Acad::ErrorStatus SetNormalizationExpression(  
    const ACHAR* pszNormalizationExpression  
);
```

Parameters

Description

pszNormalizationExpression Input normalization expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetValueToIgnore Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the values-to-ignore expression.

```
virtual Acad::ErrorStatus SetValueToIgnore(  
    const ACHAR* pszValuesToIgnore  
);
```

Parameters	Description
pszValuesToIgnore	Input values-to-ignore expression.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Update Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Builds topology stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const ACHAR* pTopoName,  
    Adesk::IntDbId lTopoElemId,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pTopoName	Input topology name.
lTopoElemId	Input topology element ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTable Class

[Classes](#)

Defines a display-management thematic table and its items.

```
class AcMapDMThematicTable : public AcDbDataTable;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTable](#) Destroys an instance of this class.



[AcMapDMThematicTable](#) Constructs an instance of this class.

[AppendRow](#) Appends a specified row to the thematic table.

[Clear](#) Empties all contents of the thematic table.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#) Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[GetCellAt](#) Opens and retrieves an annotation cell item.

[GetCellAt](#) Opens and retrieves a block cell item.

[GetCellAt](#) Opens and retrieves a color cell item.

[GetCellAt](#) Opens and retrieves a data cell item.

[GetCellAt](#) Opens and retrieves a hatch cell item.

[GetCellAt](#) Opens and retrieves a legend-text cell item.

[GetCellAt](#) Opens and retrieves a linestyle cell item.

<a href="#">GetCellAt</a>	Opens and retrieves a linetype cell item.
<a href="#">GetCellAt</a>	Opens and retrieves a lineweight cell item.
<a href="#">GetCellAt</a>	Opens and retrieves a plotstyle cell item.
<a href="#">GetCellAt</a>	Opens and retrieves a text cell item.
<a href="#">GetRowAt</a>	Opens and retrieves an entire row.
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">SetCellAt</a>	Sets the annotation style for a specified cell.
<a href="#">SetCellAt</a>	Sets the block for a specified cell.
<a href="#">SetCellAt</a>	Sets the item color for a specified cell.
<a href="#">SetCellAt</a>	Sets the value for a specified cell.
<a href="#">SetCellAt</a>	Sets the hatch for a specified cell.
<a href="#">SetCellAt</a>	Sets the legend text for a specified cell.
<a href="#">SetCellAt</a>	Sets the linestyle for a specified cell.
<a href="#">SetCellAt</a>	Sets the linetype for a specified cell.
<a href="#">SetCellAt</a>	Sets the lineweight for a specified cell.
<a href="#">SetCellAt</a>	Sets the plotstyle for a specified cell.
<a href="#">SetCellAt</a>	Sets the text for a specified cell.
<a href="#">SetRowAt</a>	Sets the row at a specified position.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemAnnotation Class

[Classes](#)

Represents an annotation cell of a thematic definition table.

```
class AcMapDMThematicTableItemAnnotation : public AcDbObject;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemAnnotation](#) Destroys an instance of this class.



[AcMapDMThematicTableItemAnnotation](#) Constructs an instance of this class.



[AcMapDMThematicTableItemAnnotation](#) Constructs an instance of this class by using another instance of this class.

[ClearColorOverride](#) Disables the color override.

[ClearLayerOverride](#) Disables the layer override.

[ClearLinetypeOverride](#) Disables the linetype override.

[ClearLineweightOverride](#) Disables the lineweight override.

[ClearRotationOverride](#) Disables the rotation override.

[ClearScaleOverride](#) Disables the scale override.

[ColorOverride](#) Retrieves the color override.

[ColorOverrideExpression](#) Retrieves the color override expression.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

<a href="#"><u>dxfInFields</u></a>	Lets this object read its data. See also <code>dxfInFields()</code> in the AutoCAD ObjectARX Developer's Guide.
<a href="#"><u>dxfOutFields</u></a>	Lets this object write its data. See also <code>dxfOutFields()</code> in the AutoCAD ObjectARX Developer's Guide.
<a href="#"><u>Implementation</u></a>	Returns the implementation object.
<a href="#"><u>IsColorOverrideEnabled</u></a>	Indicates the enable state of the color override.
<a href="#"><u>IsLayerOverrideEnabled</u></a>	Indicates the enable state of the layer override.
<a href="#"><u>IsLinetypeOverrideEnabled</u></a>	Indicates the enable state of the linetype override.
<a href="#"><u>IsLineweightOverrideEnabled</u></a>	Indicates the enable state of the lineweight override.
<a href="#"><u>IsRotationOverrideEnabled</u></a>	Indicates the enable state of the rotation override.
<a href="#"><u>IsScaleOverrideEnabled</u></a>	Indicates the enable state of the scale override.
<a href="#"><u>LayerOverride</u></a>	Retrieves the layer override.
<a href="#"><u>LayerOverrideExpression</u></a>	Retrieves the layer override expression.
<a href="#"><u>LinetypeOverride</u></a>	Retrieves the linetype override.
<a href="#"><u>LinetypeOverrideExpression</u></a>	Retrieves the linetype override expression.
<a href="#"><u>LineweightOverride</u></a>	Retrieves the lineweight override.
<a href="#"><u>LineweightOverrideExpression</u></a>	Retrieves the lineweight override expression.
<a href="#"><u>LocationOverrideExpression</u></a>	Retrieves the annotation location (insertion point) override expression.
<a href="#"><u>Read</u></a>	Lets this object read in its data from an input stream.

<a href="#"><u>RotationOverride</u></a>	Retrieves the rotation override.
<a href="#"><u>RotationOverrideExpression</u></a>	Retrieves the annotation rotation override expression.
<a href="#"><u>ScaleOverride</u></a>	Retrieves the scale override.
<a href="#"><u>ScaleOverrideExpression</u></a>	Retrieves the annotation scale override expression.
<a href="#"><u>SetColorOverride</u></a>	Sets the color override value.
<a href="#"><u>SetColorOverrideExpression</u></a>	Sets the annotation Color override expression.
<a href="#"><u>SetLayerOverride</u></a>	Sets the layer override value.
<a href="#"><u>SetLayerOverrideExpression</u></a>	Sets the annotation Layer override expression.
<a href="#"><u>SetLinetypeOverride</u></a>	Sets the linetype override value.
<a href="#"><u>SetLinetypeOverrideExpression</u></a>	Sets the annotation Linetype override expression.
<a href="#"><u>SetLineweightOverride</u></a>	Sets the lineweight override value.
<a href="#"><u>SetLineweightOverrideExpression</u></a>	Sets the annotation Lineweight override expression.
<a href="#"><u>SetLocationOverrideExpression</u></a>	Sets the annotation location (insertion point) override expression.
<a href="#"><u>SetRotationOverride</u></a>	Sets the rotation override value.
<a href="#"><u>SetRotationOverrideExpression</u></a>	Sets the annotation Rotation override expression.
<a href="#"><u>SetScaleOverride</u></a>	Sets the scale override value.
<a href="#"><u>SetScaleOverrideExpression</u></a>	Sets the annotation Scale override expression.
<a href="#"><u>SetTemplateId</u></a>	Sets the template name for this annotation alteration.
<a href="#"><u>TemplateId</u></a>	Retrieves the annotation template Id for this alteration.
<a href="#"><u>Write</u></a>	Writes object data to an output stream.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: Read Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Lets this object read in its data from an input stream.

```
void Read(  
    std::wistream& in  
);
```

Parameters

Description

in

Input filer to use to read the object's data.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetColorOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Color override expression.

```
virtual Acad::ErrorStatus SetColorOverrideExpression(  
    const ACHAR * pszColorOverrideExpression  
);
```

Parameters

Description

pszColorOverride

Input color override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetLayerOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Layer override expression.

```
virtual Acad::ErrorStatus SetLayerOverrideExpression(  
    const ACHAR * pszLayerOverrideExpression  
);
```

Parameters	Description
pszLayerOverride	Input layer override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetLinetypeOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Linetype override expression.

```
virtual Acad::ErrorStatus SetLinetypeOverrideExpression(  
    const ACHAR * pszLinetypeOverrideExpression  
);
```

Parameters	Description
pszLinetypeOverride	Input linetype override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetLineweightOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Lineweight override expression.

```
virtual Acad::ErrorStatus SetLineweightOverrideExpression(  
    const ACHAR * pszLineweightOverrideExpression  
);
```

Parameters

Description

pszLineweightOverride Input lineweight override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetLocationOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation location (insertion point) override expression.

```
virtual Acad::ErrorStatus SetLocationOverrideExpression(  
    const ACHAR * pszLocationOverrideExpression  
);
```

Parameters	Description
pszLocationOverride	Input location override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetRotationOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Rotation override expression.

```
virtual Acad::ErrorStatus SetRotationOverrideExpression(  
    const ACHAR * pszRotationOverrideExpression  
);
```

Parameters

Description

pszRotationOverride

Input rotation override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

SetScaleOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the annotation Scale override expression.

```
virtual Acad::ErrorStatus SetScaleOverrideExpression(  
    const ACHAR * pszScaleOverrideExpression  
);
```

Parameters

Description

pszScaleOverride

Input scale override expression value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: Write Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Writes object data to an output stream.

```
void write(  
    std::wostream& o  
) const;
```

Parameters

Description

o

Input stream filer to use to write in the object's data.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemBlock Class

[Classes](#)

Represents a block cell of a thematic definition table.

```
class AcMapDMThematicTableItemBlock : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemBlock](#) Destroys an instance of this class.



[AcMapDMThematicTableItemBlock](#) Constructs an instance of this class.



[AcMapDMThematicTableItemBlock](#) Constructs an instance of this class by using another instance of this class.

[Angle](#) Retrieves the angle.

[BlockId](#) Retrieves the block AcDbObjectId.

[BlockName](#) Retrieves the block name.

[dwgInFields](#) Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#) Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#) Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#) Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#) Returns the implementation object.

[LayerId](#) Retrieves the layer AcDbObjectId.

[LayerName](#) Retrieves the layer name.

<a href="#">Scale</a>	Retrieves the scale.
<a href="#">SetAngle</a>	Sets the angle.
<a href="#">SetBlockId</a>	Sets the block by AcDbObjectId.
<a href="#">SetBlockName</a>	Sets the block name.
<a href="#">SetLayerId</a>	Sets the layer by AcDbObjectId.
<a href="#">SetLayerName</a>	Sets the layer name.
<a href="#">SetScale</a>	Sets the scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetBlockName Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the block name.

```
virtual Acad::ErrorStatus SetBlockName(  
    const ACHAR* pszBlockName  
);
```

Parameters	Description
pszBlockName	Input block name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetLayerName Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the layer name.

```
virtual Acad::ErrorStatus SetLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemColor Class

[Classes](#)

Represents a color cell of a thematic definition table.

```
class AcMapDMThematicTableItemColor : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemColor](#) Destroys an instance of this class.



[AcMapDMThematicTableItemColor](#) Constructs an instance of this class.



[AcMapDMThematicTableItemColor](#) Constructs an instance of this class by using another instance of this class.

[Color](#) Retrieves the color.

[dwgInFields](#) Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#) Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#) Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#) Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#) Returns the implementation object.

[SetColor](#) Sets the color.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemDataValue Class

[Classes](#)

Represents a text/value-pair cell of a thematic definition table.

```
class AcMapDMThematicTableItemDataValue : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemDataValue](#) Destroys an instance of this class.



[AcMapDMThematicTableItemDataValue](#) Constructs an instance of this class.



[AcMapDMThematicTableItemDataValue](#) Constructs an instance of this class by using another instance of this class.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#)

Returns the implementation object.

[SetText](#)

Sets the text.

[SetValue](#)

Sets the value.

[Text](#)

Retrieves the text.

[Value](#)

Retrieves the value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: SetText Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Sets the text.

```
virtual Acad::ErrorStatus SetText(  
    const ACHAR* pText  
);
```

Parameters	Description
------------	-------------

pText	Input text.
-------	-------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemHatch Class

[Classes](#)

Represents a hatch cell of a thematic definition table.

```
class AcMapDMThematicTableItemHatch : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemHatch](#) Destroys an instance of this class.



[AcMapDMThematicTableItemHatch](#) Constructs an instance of this class.



[AcMapDMThematicTableItemHatch](#) Constructs an instance of this class by using another instance of this class.

[Angle](#) Retrieves the angle.

[Color](#) Retrieves the color.

[dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#) Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#) Returns the implementation object.

[LayerId](#) Retrieves the layer `AcDbObjectId`.

[LayerName](#) Retrieves the layer name.

[PatternName](#) Retrieves the pattern name.

<a href="#">Scale</a>	Retrieves the scale.
<a href="#">SetAngle</a>	Sets the angle.
<a href="#">SetColor</a>	Sets the color.
<a href="#">SetLayerId</a>	Sets the layer AcDbObjectId.
<a href="#">SetLayerName</a>	Sets the layer name.
<a href="#">SetPatternName</a>	Sets the pattern name.
<a href="#">SetScale</a>	Sets the scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetLayerName Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the layer name.

```
virtual Acad::ErrorStatus SetLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetPatternName Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the pattern name.

```
virtual Acad::ErrorStatus SetPatternName(  
    const ACHAR* pszPatternName  
);
```

Parameters	Description
pszPatternName	Input pattern name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemLegendText Class

[Classes](#)

Represents a legend text cell of a thematic definition table.

```
class AcMapDMThematicTableItemLegendText : public AcDbObject;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemLegendText](#) Destroys an instance of this class.



[AcMapDMThematicTableItemLegendText](#) Constructs an instance of this class.



[AcMapDMThematicTableItemLegendText](#) Constructs an instance of this class by using another instance of this class.

[Color](#)

Retrieves the color of the cell.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[Height](#)

Retrieves the height of the cell.

[Implementation](#)

Returns the implementation object.

[LayerId](#)

Retrieves the layer AcDbObjectId of the cell.

[LayerName](#)

Retrieves the layer name of the cell.

[SetColor](#)

Sets the color of the cell.

[SetHeight](#)

Sets the height of the cell.

[SetLayerId](#)

Sets the layer AcDbObjectId of the cell.

[SetLayerName](#)

Sets the layer name of the cell.

[SetStyleId](#)

Sets the style AcDbObjectId of the cell.

[SetStyleName](#)

Sets the style name of the cell.

[SetText](#)

Sets the text of the cell.

[StyleId](#)

Retrieves the style AcDbObjectId of the cell.

[StyleName](#)

Retrieves the style name of the cell.

[Text](#)

Retrieves the text of the cell.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: SetLayerName Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the layer name of the cell.

```
virtual Acad::ErrorStatus SetLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: SetStyleName Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the style name of the cell.

```
virtual Acad::ErrorStatus SetStyleName(  
    const ACHAR* pszStyleName  
);
```

Parameters	Description
pszStyleName	Input style name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText::SetText Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the text of the cell.

```
virtual Acad::ErrorStatus SetText(  
    const ACHAR* pszText  
);
```

Parameters	Description
------------	-------------

pszText	Input text.
---------	-------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemLinestyle Class

[Classes](#)

Represents a linestyle cell of a thematic definition table.

```
class AcMapDMThematicTableItemLinestyle : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemLinestyle](#) Destroys an instance of this class.



[AcMapDMThematicTableItemLinestyle](#) Constructs an instance of this class.



[AcMapDMThematicTableItemLinestyle](#) Constructs an instance of this class by using another instance of this class.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#)

Returns the implementation object.

[LayerId](#)

Retrieves the layer `AcDbObjectId`.

[LayerName](#)

Retrieves the layer name.

[LinetypeId](#)

Retrieves the linetype `AcDbObjectId`.

<a href="#">LinetypeName</a>	Retrieves the linetype name.
<a href="#">SetLayerId</a>	Sets the layer AcDbObjectId.
<a href="#">SetLayerName</a>	Sets the layer name.
<a href="#">SetLinetypeId</a>	Sets the linetype AcDbObjectId.
<a href="#">SetLinetypeName</a>	Sets the linetype name.
<a href="#">SetWidth</a>	Sets the width.
<a href="#">Width</a>	Retrieves the width.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: SetLayerName Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Sets the layer name.

```
virtual Acad::ErrorStatus SetLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: SetLinetypeName Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Sets the linetype name.

```
virtual Acad::ErrorStatus SetLinetypeName(  
    const ACHAR* pszLinetypeName  
);
```

Parameters	Description
pszLinetypeName	Input linetype name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemLinetype Class

[Classes](#)

Represents a linetype cell of a thematic definition table.

```
class AcMapDMThematicTableItemLinetype : public AcDbObject;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemLinetype](#) Destroys an instance of this class.



[AcMapDMThematicTableItemLinetype](#) Constructs an instance of this class.



[AcMapDMThematicTableItemLinetype](#) Constructs an instance of this class by using another instance of this class.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#)

Returns the implementation object.

[Linetype](#)

Retrieves the linetype.

[LinetypeId](#)

Retrieves the linetype `AcDbObjectId`.

[SetLinetype](#)

Sets the linetype name.

## [SetLinetypeId](#)

Sets the linetype AcDbObjectId.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: SetLinetype Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Sets the linetype name.

```
virtual Acad::ErrorStatus SetLinetype(  
    const ACHAR * pszLinetype  
);
```

Parameters	Description
pszLinetype	Input linetype name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemLineweight Class

[Classes](#)

Class representing lineweight cell of thematic definition table.

```
class AcMapDMThematicTableItemLineweight : public AcDbObject;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemLineweight](#)

Destroys an instance of this class.



[AcMapDMThematicTableItemLineweight](#)

Constructs an instance of this class.



[AcMapDMThematicTableItemLineweight](#)

Constructs an instance of this class by using another instance of this class.

[dwgInFields](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#)

Returns the implementation object.

[Lineweight](#)

Retrieves the lineweight.

[SetLineweight](#)

Sets the lineweight.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemPlotstyle Class

[Classes](#)

Represents a plotstyle cell of a thematic definition table.

```
class AcMapDMThematicTableItemPlotstyle : public AcDbObject;  
File
```

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemPlotstyle](#) Destroys an instance of this class.



[AcMapDMThematicTableItemPlotstyle](#) Constructs an instance of this class.



[AcMapDMThematicTableItemPlotstyle](#) Constructs an instance of this class by using another instance of this class.

[dwgInFields](#)

Lets this object read its data. See also [dwgInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also [dwgOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also [dxfInFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also [dxfOutFields\(\)](#) in the AutoCAD ObjectARX Developer's Guide.

[Implementation](#)

Returns the implementation object.

[PlotstyleName](#)

Retrieves the plotstyle name.

[SetPlotstyleName](#)

Sets the plotstyle name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: SetPlotstyleName Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Sets the plotstyle name.

```
virtual Acad::ErrorStatus SetPlotstyleName(  
    const ACHAR* pszPlotstyleName  
);
```

Parameters	Description
pszPlotstyleName	Input plotstyle name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableItemText Class

[Classes](#)

Represents a text cell of a thematic definition table.

```
class AcMapDMThematicTableItemText : public AcDbObject;
```

File

DmThematicTable.h

☐ Methods



[~AcMapDMThematicTableItemText](#)

Destroys an instance of this class.



[AcMapDMThematicTableItemText](#)

Constructs an instance of this class.



[AcMapDMThematicTableItemText](#)

Constructs an instance of this class by using another instance of this class.

[Angle](#)

Retrieves the angle.

[Color](#)

Retrieves the color.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide.

[Height](#)

Retrieves the height.

[Implementation](#)

Returns the implementation object.

[InsertPoint](#)

Retrieves the insertion point.

[Justification](#)

Retrieves the justification.

<a href="#">LayerId</a>	Retrieves the layer AcDbObjectId.
<a href="#">LayerName</a>	Retrieves the layer name.
<a href="#">SetAngle</a>	Sets the angle.
<a href="#">SetColor</a>	Sets the color.
<a href="#">SetHeight</a>	Sets the height.
<a href="#">SetInsertPoint</a>	Sets the insertion point.
<a href="#">SetJustification</a>	Sets the justification type.
<a href="#">SetLayerId</a>	Sets the layer AcDbObjectId.
<a href="#">SetLayerName</a>	Sets the layer name.
<a href="#">SetStyleId</a>	Sets the style AcDbObjectId.
<a href="#">SetStyleName</a>	Sets the style name.
<a href="#">SetText</a>	Sets the text.
<a href="#">StyleId</a>	Retrieves the style AcDbObjectId.
<a href="#">StyleName</a>	Retrieves the style name.
<a href="#">Text</a>	Retrieves the text.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetLayerName Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the layer name.

```
virtual Acad::ErrorStatus SetLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetStyleName Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the style name.

```
virtual Acad::ErrorStatus SetStyleName(  
    const ACHAR* pszStyleName  
);
```

Parameters	Description
pszStyleName	Input style name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetText Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the text.

```
virtual Acad::ErrorStatus SetText(  
    const ACHAR* pszText  
);
```

Parameters	Description
pszText	Input the text.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMThematicTableRow Class

[Classes](#)

Utility class that defines a table row and allows easier access to its column items.

```
class AcMapDMThematicTableRow;
```

File

DmThematicTable.h

☐ Data Members

- [mpAnnotation](#) Annotation value type.
- [mpBlock](#) Block value type.
- [mpColor](#) Color value type.
- [mpDataValue](#) Text/value-pair value type.
- [mpHatch](#) Hatch value type.
- [mpLegendText](#) Legend-text value type.
- [mpLinestyle](#) Linestyle value type.
- [mpLinetype](#) Linetype value type.
- [mpLineweight](#) Lineweight value type.
- [mpPlotstyle](#) Plotstyle value type.
- [mpText](#) Text value type.

☐ Methods

-  [~AcMapDMThematicTableRow](#) Destroys an instance of this class.
-  [AcMapDMThematicTableRow](#) Constructs an instance of this class.
- [closeRow](#) Closes all members.
- [Init](#) Initializes the member variables.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMTopoElement Class

[Classes](#)

Defines a query for all topological elements in the current drawing.

```
class AcMapDMTopoElement : public AcMapDMTopoQueryElement;  
File
```

DmTopoElement.h

☐ Methods

⇒ [~AcMapDMTopoElement](#)

Destroys an instance of this class.

⇒ [AcMapDMTopoElement](#)

Constructs an instance of this class.

[AcquireEntities](#)

Runs the topological query against the current drawing and acquires both selected entities and topology objects.

[ClonesObjectsFromExternalSource](#)

Clones objects from the source drawings.

[DismissEntities](#)

Erases queried objects.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxInFields](#)

Lets this object read its data. See also `dxInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it

directly.

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[EvaluateExpressionValues](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Invoked when an AutoCAD Map project is initialized.

[OnMapProjectInitialized](#)

Sets the query definition's data-source descriptor.

[SetAcquisitionCriteria](#)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)  
AcMapDMTopoElement:: EvaluateExpressionValues Method  
[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)  
AcMapDMTopoElement:: EvaluateExpressionValues Method  
[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

[AcMapDMTopoElement:: GetAcquisitionCriteria Method](#)

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters

Description

pszString

Output pointer to the data-source descriptor, as a string representation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMTopoElementDataSourceDescriptor Class

[Classes](#)

Defines a topology data-source.

```
class AcMapDMTopoElementDataSourceDescriptor : public AcMapDMDataSou  
File
```

DmTopoElement.h

☐ Methods



[~AcMapDMTopoElementDataSourceDescriptor](#)

Destroys an instance of this class.



[AcMapDMTopoElementDataSourceDescriptor](#)

Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[GetTopologyName](#)

Retrieves the topology name.

[SetAcquisitionStatement](#)

Sets the query string.

[SetTopologyName](#)

Sets the topology name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor::

GetAcquisitionStatement Method

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor:: GetTopologyName Method

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Retrieves the topology name.

```
virtual Acad::ErrorStatus GetTopologyName(  
    const ACHAR*& pszTopologyName  
) const;
```

Parameters	Description
pszTopologyName	Output topology name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor::

SetAcquisitionStatement Method

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor:: SetTopologyName Method

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Sets the topology name.

```
virtual Acad::ErrorStatus SetTopologyName(  
    const ACHAR* pszTopologyName  
);
```

Parameters	Description
pszTopologyName	Input name of the topology.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMTopoQueryDataSourceDescriptor Class

[Classes](#)

Defines a topology data-source.

```
class AcMapDMTopoQueryDataSourceDescriptor : public AcMapDMAttachedD  
File
```

DmTopoQueryElement.h

☐ Methods



[~AcMapDMTopoQueryDataSourceDescriptor](#)

Destroys an instance of this class.



[AcMapDMTopoQueryDataSourceDescriptor](#)

Constructs an instance of this class.

[GetAcquisitionStatement](#)

Retrieves the query's string representation.

[GetQuery](#)

Retrieves the query definition.

[GetTopologyName](#)

Retrieves the topology name.

[SetAcquisitionStatement](#)

Sets the query string.

[SetQuery](#)

Sets the query definition.

[SetTopologyName](#)

Sets the topology name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: GetAcquisitionStatement Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Retrieves the query's string representation.

```
virtual Acad::ErrorStatus GetAcquisitionStatement(  
    ACHAR*& pszStatement  
) const;
```

Parameters	Description
pszStatement	Output query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: GetTopologyName Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Retrieves the topology name.

```
virtual Acad::ErrorStatus GetTopologyName(  
    const ACHAR*& pszTopologyName  
) const;
```

Parameters	Description
pszTopologyName	Output topology name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: SetAcquisitionStatement Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Sets the query string.

```
virtual Acad::ErrorStatus SetAcquisitionStatement(  
    const ACHAR* pszStatement  
);
```

Parameters	Description
pszStatement	Input query string.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: SetTopologyName Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Sets the topology name.

```
virtual Acad::ErrorStatus SetTopologyName(  
    const ACHAR* pszTopologyName  
);
```

Parameters

Description

pszTopologyName

Input name of the topology.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapDMTopoQueryElement Class

[Classes](#)

Defines a query element that acquires topology objects in the source drawing.

```
class AcMapDMTopoQueryElement : public AcMapDMAttachedDwgsQueryElement  
File
```

DmTopoQueryElement.h

☐ Methods

- ◆ [~AcMapDMTopoQueryElement](#) Destroys an instance of this class.
- ◆ [AcMapDMTopoQueryElement](#) Constructs an instance of this class.
- [AcquireEntities](#) Runs the topological query and acquires both selected entities and topology objects.
- [AddStyle](#) Adds a new style to this element.
- [AddStyle](#) Adds a new style to this element.
- [ClonesObjectsFromExternalSource](#) Clones objects from the source drawings.
- [DismissEntities](#) Erases entities that are part of this query element.
- [DismissStylization](#) Dismisses the current stylization.
- [dwgInFields](#) Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
- [dwgOutFields](#) Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.
- [dxfInFields](#) Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX

[dxfInFields](#)

Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

[EmptyCookieJar](#)

Called during database destruction to trigger the deletion of topology-managed cookies.

[EnableStyle](#)

Enables or disables a style reference.

[EvaluateExpressionValues](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

[EvaluateExpressionValues](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

[GetAcquisitionCriteria](#)

Retrieves a query definition's data-source descriptor.

[GetAcquisitionCriteria](#)

Retrieves the query definition's data-source descriptor.

[GetTopologyName](#)

Retrieves the topology name.

[OnMapProjectInitialized](#)

Invoked when an AutoCAD Map project is initialized.

[RemoveStyle](#)

Removes a style reference from this element.

[SetAcquisitionCriteria](#)

Sets the query definition's data-source descriptor.

[SetTopologyName](#)

Sets the name of the topology.

[setVisible](#)

Makes the entities that are part of this element visible or invisible.

[UpdateStylization](#)

Stylizes the current or updated selection at the current scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: AddStyle Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Adds a new style to this element.

```
virtual Acad::ErrorStatus AddStyle(  
    AcDbObjectId& styleRefId,  
    AcMapDMStyle* pStyle,  
    AcMapDMAllStyleReferencesIterator& Position  
);
```

Parameters	Description
styleRefId	Output ID of the added style reference.
pStyle	Input pointer to a AcMapDMStyleobject.
Position	Input position at which to add the style. See also <a href="#">AcMapDMAllStyleReferencesIterator</a>

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The style object and the owning AcMapDMMap object must be closed for this function to succeed. TopoElement will not accept [AcMapDMSEAnnotationStyle](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: AddStyle Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Adds a new style to this element.

```
virtual Acad::ErrorStatus AddStyle(  
    AcDbObjectId& styleRefId,  
    const AcDbObjectId& styleId,  
    AcMapDMAllStyleReferencesIterator& Position  
);
```

Parameters	Description
styleRefId	Output ID of the added style reference.
styleId	Input ID of the style.
Position	Input position at which to add the style. See also <a href="#">AcMapDMAllStyleReferencesIterator</a>

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The style object and the owning AcMapDMMap object must be closed for this function to succeed. TopoElement will not accept [AcMapDMSEAnnotationStyle](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: EvaluateExpressionValues Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Evaluates two expressions against the objects that meet the acquisition criteria.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues1,  
    AcArray<AcMapValue*>& arValues2,  
    const ACHAR* kpszExpression1,  
    AcMap::EDataType kDataType1,  
    const ACHAR* kpszExpression2,  
    AcMap::EDataType kDataType2  
);
```

Parameters	Description
arValues1	Output array of evaluated values.
arValues2	Output array of evaluated values.
kpszExpression1	Input expression to evaluate.
kDataType1	Input return data type expected.
kpszExpression2	Input expression to evaluate.
kDataType2	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

In the case of query elements, objects are not cloned and the expressions are evaluated against the original objects. If the expressions are not applicable to an entity (the .AREA for a line, for example), the result for this entity has type AcMap::kUnknownType.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: EvaluateExpressionValues Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Evaluates expression(s) against the element selection and returns values for the selected entities.

```
virtual Acad::ErrorStatus EvaluateExpressionValues(  
    AcArray<AcMapValue*>& arValues,  
    const ACHAR* kpszExpression,  
    AcMap::EDataType kDataType  
);
```

Parameters	Description
arValues	Output array of evaluated values.
kpszExpression	Input expression to evaluate.
kDataType	Input return data type expected.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If entities have not yet been acquired, this function acquires them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: GetAcquisitionCriteria Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Retrieves a query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    ACHAR*& pszString  
) const;
```

Parameters	Description
pszString	Output pointer to the data-source descriptor, as a string representation.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: GetTopologyName Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Retrieves the topology name.

```
Acad::ErrorStatus GetTopologyName(  
    const ACHAR*& kpszName  
) const;
```

Parameters	Description
kpszName	Output topology name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: SetTopologyName Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Sets the name of the topology.

```
Acad::ErrorStatus SetTopologyName(  
    const ACHAR* kpszName  
);
```

Parameters	Description
kpszName	Input topology name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Current-drawing topologies are ignored.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Import-Export

### ▫ Namespaces

#### [AcMapIE](#)

This namespace provides structs and enumerations that the AcMapIE classes use in various import-export contexts.

### ▫ Functions

#### [AcMapExporter](#)

Retrieves the AutoCAD Map exporter object, as a singleton.

#### [AcMapExportFormatIterator](#)

Retrieves the AutoCAD Map export format iterator.

#### [AcMapImporter](#)

Retrieves the AutoCAD Map importer object, as a singleton.

#### [AcMapImportFormatIterator](#)

Retrieves the AutoCAD Map import format iterator.

### ▫ Classes

#### [AcMapIEColumn](#)

Provides functions that handle imported columns (also called fields).

#### [AcMapIEColumnIterator](#)

An iterator over a collection of

#### [AcMapIEExporter](#)

Provides functions that handle the export process.

Provides callback functions for export reactors. To implement an export reactor, derive classes from these virtual base classes, overwriting the desired functions. Pass instances of the classes to the exporter (an instance of AcMapIEExporter) and the exporter will call back into your classes at various points in the export process. This behavior allows you to modify the export process at runtime or track the export process externally. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

#### [AcMapIEExportReactor](#)

#### [AcMapIEExpressionTargetIterator](#)

An iterator over a collection of expression-target pairs.

Provides functions that handle import and

[AcMapIEFormat](#)

export file formats.

[AcMapIEFormatIterator](#)

An iterator over a collection of

[AcMapIEImporter](#)

Provides functions that handle the import process.

[AcMapIEImportReactor](#)

Provides callback functions for import reactors.

[AcMapIEInputLayer](#)

Provides functions that handle input layers (also called themes).

[AcMapIEInputLayerIterator](#)

An iterator over a collection of

[AcMapIENameValueIterator](#)

An iterator over a collection of name-value pairs.

## Links

[Import-Export](#), [Structures](#)

AcMapIE Namespace

[Import-Export](#) | [Structures](#)

This namespace provides structs and enumerations that the AcMapIE classes use in various import-export contexts. For more information, search for *importing* and *exporting* in AutoCAD Map Help.

### ▣ Enumerations

-  [ErrCode](#) Enumerates the import-export error codes.
-  [DataType](#) This is record AcMapIE::DataType.
-  [GeometryType](#) Enumerates the geometric types to export.
-  [ImportDataMapping](#) Enumerates the types of incoming-column data mappings.
-  [ExportClassMapping](#) Enumerates the type of source data for mapping from source to target Feature classes.
-  [LayerNameType](#) Enumerates the ways to use layer names in the importer.
-  [LocationOption](#) Enumerates the types of location window.
-  [PointMappingType](#) Enumerates the ways to import points.
-  [StorageType](#) Enumerates the storage types of import and export formats. These enumerated values are returned by functions in the AcMapIEFormatclass and describe the data set type. A value can apply to only import, only export, or to both.

### ▣ Structures

-  [CExportResults](#) Returns the results of an export.
-  [CImportResults](#) Returns the results of an import.
-  [CNewExportResults](#) Returns the results of an export.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

CExportResults Structure

[AcMapIE Namespace](#)

Returns the results of an export.

```
struct CExportResults {  
    unsigned long m\_ulEntitiesExported;  
    unsigned long m\_ulEntitiesSkippedCouldntTransform;  
};  
File
```

AcMapIEEnum.h

Parameters	Description
<a href="#">m_ulEntitiesExported</a>	Total number of entities that were exported. This will correspond to the number we thought we were exporting.
<a href="#">m_ulEntitiesSkippedCouldntTransform</a>	Number of entities that were not exported because they could not be transformed to the target coordinate system.

Remarks

Use this struct to monitor an export or to summarize an export for an end-user, warning the user about export anomalies if necessary.

▣ Data Members

<a href="#">m_ulEntitiesExported</a>	This is <a href="#">m_ulEntitiesExported</a> , a member of class CExportResults.
<a href="#">m_ulEntitiesSkippedCouldntTransform</a>	This is <a href="#">m_ulEntitiesSkippedCouldntTransform</a> , a member of class CExportResults.

▣ Methods

◆ <a href="#">CExportResults</a>	This is CExportResults, a member of class CExportResults.
----------------------------------	---

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

CImportResults Structure

[AcMapIE Namespace](#)

Returns the results of an import.

```
struct CImportResults {  
    unsigned long m\_ulEntitiesImported;  
    unsigned long m\_ulEntitiesSkippedCouldntTransform;  
    bool m\_bEvaluateHatchFailed;  
    unsigned long m\_ulEntitiesWithColorCloseToBackground;  
    unsigned long m\_ulEntitiesClassified;  
    unsigned long m\_ulEntitiesNotClassifiable;  
    unsigned long m\_ulOutOfRangeNotFixed;  
};
```

File

AcMapIEEnum.h

Parameters

Description

`m_ulEntitiesImported`

Total number of entities that were imported.

`m_ulEntitiesSkippedCouldntTransform`

Number of entities that were not imported because they could not be transformed to the target coordinate system.

`m_bEvaluateHatchFailed`

Number of imported entities whose hatch patterns appear incorrectly on the drawing. If the scale of the global hatch pattern is incorrect for the drawing, then AutoCAD Map will not draw the hatch pattern and will display empty polygons. This situation will not occur if the global hatch pattern is solid (the AutoCAD Map default).

Number of imported entities whose color is almost indistinguishable

m_ulEntitiesWithColorCloseToBackground	from the background color and therefore appear invisible on the drawing. This situation will not occur if the file <i>MapImport.ini</i> contains the setting "Color=Closest ACI only", which forces AutoCAD Map to adjust colors so that entities are visible.
m_ulEntitiesClassified	Number of imported entities that were tagged with a feature class.
m_ulEntitiesNotClassifiable	Number of imported entities that could not be tagged with a feature class.
m_ulOutOfRangeNotFixed	Number of imported entities with at least one out-of-range property that could not be reset to its default value. Often, a property cannot be reset because a line type is not loaded or a layer does not exist. Note that this value counts the number of entities, not the number of properties; it is possible for an entity to have multiple out-of-range properties that could not be reset.

## Remarks

Use this struct to monitor an import or to summarize an import for an end-user, warning the user about import anomalies if necessary.

## ▣ Data Members

[m\\_bEvaluateHatchFailed](#)

This is m\_bEvaluateHatchFailed, a member of class CImportResults.

[m\\_ulEntitiesClassified](#)

This is m\_ulEntitiesClassified, a member of class CImportResults.

[m\\_ulEntitiesImported](#)

This is m\_ulEntitiesImported, a member of class CImportResults.

This is m\_ulEntitiesNotClassifiable,

[m\\_ulEntitiesNotClassifiable](#)

member of class CImportResults.

[m\\_ulEntitiesSkippedCouldntTransform](#)

This is  
m\_ulEntitiesSkippedCouldntTransf  
member of class CImportResults.

[m\\_ulEntitiesWithColorCloseToBackground](#)

This is  
m\_ulEntitiesWithColorCloseToBacl  
a member of class CImportResults.

[m\\_ulOutOfRangeNotFixed](#)

This is m\_ulOutOfRangeNotFixed,  
member of class CImportResults.

#### ☐ Methods

◆ [CImportResults](#)

This is CImportResults, a member of class  
CImportResults.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

CNewExportResults Structure

[AcMapIE Namespace](#)

Returns the results of an export.

```
struct CNewExportResults : public CExportResults {  
    unsigned long m\_ulTotalEntitiesExported;  
    unsigned long m\_ulEntitiesExportedMoreThanOnce;  
};  
File
```

AcMapIEEnum.h

Parameters	Description
<a href="#">m_ulEntitiesExported</a>	Total number of entities that were exported. This will correspond to the number we thought we were exporting.
<a href="#">m_ulEntitiesSkippedCouldntTransform</a>	Number of entities that were not exported because they could not be transformed to the target coordinate system.
<a href="#">m_ulTotalEntitiesExported</a>	With multi class export some entities will be exported more than once. This contains the total count.
<a href="#">m_ulEntitiesExportedMoreThanOnce</a>	With multi class export, some entities will be exported more than once. This contains the count of entities which were exported multiple times.

Remarks

Use this struct to monitor an export or to summarize an export for an end-user, warning the user about export anomalies if necessary. Using CNewExportResults with Map 2007 will return additional results of total entities and entities exported multiple times. Using CNewExportResults with Map 2006 and below will always return 0 for these parameters.

## ☐ Data Members

[m\\_ulEntitiesExportedMoreThanOnce](#) This is m\_ulEntitiesExportedMoreThanOnce, a member of class CNewExportResults.

[m\\_ulTotalEntitiesExported](#) This is m\_ulTotalEntitiesExported, a member of class CNewExportResults.

## ☐ Methods

◆ [CNewExportResults](#) This is CNewExportResults, a member of class CNewExportResults.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CNewExportResults Structure](#), [CNewExportResults Structure](#), [AcMapIE Namespace](#)

CNewExportResults:: m\_ulEntitiesExportedMoreThanOnce Data Member  
[CNewExportResults Structure](#) | [CNewExportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesExportedMoreThanOnce, a member of class CNewExportResults.

```
unsigned long m_ulEntitiesExportedMoreThanOnce;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CNewExportResults Structure](#), [CNewExportResults Structure](#), [AcMapIE Namespace](#)

CNewExportResults:: m\_ulTotalEntitiesExported Data Member

[CNewExportResults Structure](#) | [CNewExportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulTotalEntitiesExported, a member of class CNewExportResults.

```
unsigned long m_ulTotalEntitiesExported;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CNewExportResults Structure](#), [CNewExportResults Structure](#), [AcMapIE Namespace](#)

CNewExportResults:: CNewExportResults Constructor

[CNewExportResults Structure](#) | [CNewExportResults Structure](#) | [AcMapIE Namespace](#)

This is CNewExportResults, a member of class CNewExportResults.

```
CNewExportResults();
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: ErrorCode Enumeration

[AcMapIE Namespace](#)

Enumerates the import-export error codes.

```
enum ErrorCode {
    kErr_Invalid = 0,
    kErr_OK,
    kErr_Fail,
    kErr_OutOfMemory,
    kErr_FileError,
    kErr_DialogNotAvailable,
    kErr_TableNotSet,
    kErr_ClassNotSet,
    kErr_BadFieldName,
    kErr_BadTableName,
    kErr_ConflictWithTarget,
    kErr_Conflict,
    kErr_ConflictWithKey,
    kErr_NotInitialized,
    kErr_BadParams,
    kErr_InvalidWindow,
    kErr_InvalidMapping,
    kErr_NoBlocksInDrawing,
    kErr_CannotModifySchema,
    kErr_ClassExists,
    kErr_InvalidDataStore
};
```

File

AcMapIEEnum.h

Parameters	Description
kErr_Invalid	An invalid reference or input value.
kErr_OK	The action completed successfully.
kErr_Fail	The action failed.
kErr_OutOfMemory	Insufficient system resources.
kErr_FileError	A file-related error such as an invalid file or directory name, a missing file, or insufficient file permissions.

kErr_DialogNotAvailable	The specified dialog box is unavailable.
kErr_TableNotSet	No table was set.
kErr_ClassNotSet	No feature class was set.
kErr_BadFieldName	Column does not exist.
kErr_BadTableName	Table does not exist or already exists.
kErr_ConflictWithTarget	Target column already exists.
kErr_Conflict	Two incoming columns are mapped to the same target column.
kErr_ConflictWithKey	Target column has the same name as the unique key column.
kErr_NotInitialized	The importer or exporter is not initialized.
kErr_BadParams	Invalid arguments.
kErr_InvalidWindow	Invalid location window.
kErr_InvalidMapping	Reserved.
kErr_NoBlocksInDrawing	Unable to map points to blocks.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: DataType Enumeration

[AcMapIE Namespace](#)

This is record AcMapIE::DataType.

```
enum DataType {  
    kData_String,  
    kData_Decimal,  
    kData_Float,  
    kData_Double,  
    kData_Short,  
    kData_Long,  
    kData_Boolean,  
    kData_Date,  
    kData_Unknown,  
    kData_LongLong,  
    kData_Raster,  
    kData_Invalid  
};  
File
```

AcMapIEEnum.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: GeometryType Enumeration

[AcMapIE Namespace](#)

Enumerates the geometric types to export.

```
enum GeometryType {  
    kGeom_Invalid,  
    kGeom_All,  
    kGeom_Point,  
    kGeom_Line,  
    kGeom_Polygon,  
    kGeom_Text,  
    kGeom_Point_Line,  
    kGeom_Point_Polygon,  
    kGeom_Line_Polygon  
};
```

File

AcMapIEEnum.h

Parameters	Description
kGeom_Invalid	Invalid geometric type.
kGeom_All	Export all geometric types.
kGeom_Point	Export points (AcDbPoint and AcDbBlockReference).
kGeom_Line	Export lines (AcDbPolyline, AcDb2dPolyline, AcDbSpline, AcDb3dPolyline, and AcDbArc).
kGeom_Polygon	Export polygons (AcDbPolygon, AcDbEllipse, AcDbCircle, AcDbPolyline if closed, AcDb2dPolyline if closed, AcDbSpline if closed, and AcDb3dPolyline if closed).
kGeom_Text	Export text (AcDbText and AcDbMText).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: ImportDataMapping Enumeration

[AcMapIE Namespace](#)

Enumerates the types of incoming-column data mappings.

```
enum ImportDataMapping {  
    kNoImportMapping,  
    kNewObjectDataOnly,  
    kExistingObjectDataOnly,  
    kLinkTemplate,  
    kLinkOnly  
};
```

File

AcMapIEEnum.h

Parameters	Description
kNoImportMapping	Do not import column.
kNewObjectDataOnly	Import column to a new object-data table.
kExistingObjectDataOnly	Import column to an existing object-data table.
kLinkTemplate	Import column to an SQL table.
kLinkOnly	Import column as a link only. For a column, indicates that the column is a key column. For <a href="#">AcMapIEInputLayer::SetDataMapping()</a> , indicates that only key columns are mapped.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: ExportClassMapping Enumeration

[AcMapIE Namespace](#)

Enumerates the type of source data for mapping from source to target Feature classes.

```
enum ExportClassMapping {  
    kExport_InvalidMapping,  
    kExport_SingleClass,  
    kExport_ByLayer,  
    kExport_ByObjectClass,  
    kExport_ByObjectData,  
    kExport_ByLinkTemplate  
};  
File
```

AcMapIEEnum.h

Parameters	Description
kExport_InvalidMapping	Value not set
kExport_SingleClass	All source data is Exported to a single target Class
kExport_ByLayer	Source of each Feature Class is an AutoCAD Layer
kExport_ByObjectClass	Source of each Feature Class is an AutoCAD Map Object Class
kExport_ByObjectData	Source of each Feature Class is an AutoCAD Map Object Data Table
kExport_ByLinkTemplate	Source of each Feature Class is an AutoCAD Map Link Template

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: LayerNameType Enumeration

[AcMapIE Namespace](#)

Enumerates the ways to use layer names in the importer.

```
enum LayerNameType {  
    kLayerName_Direct,  
    kLayerName_Indirect  
};  
File
```

AcMapIEEnum.h

Parameters	Description
kLayerName_Direct	Use the specified layer name.
kLayerName_Indirect	Place the entities from this input layer on the layer specified attribute data from this column.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: LocationOption Enumeration

[AcMapIE Namespace](#)

Enumerates the types of location window.

```
enum LocationOption {  
    kDontUse,  
    kUseScreenBoundary,  
    kUseLocationWindow  
};  
File
```

AcMapIEEnum.h

Parameters	Description
kDontUse	Do not use the location window (that is, import the entire data set).
kUseScreenBoundary	Use the dimensions of the current window.
kUseLocationWindow	Use the specified window.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: PointMappingType Enumeration

[AcMapIE Namespace](#)

Enumerates the ways to import points.

```
enum PointMappingType {  
    kMapPoint_Invalid,  
    kMapPoint_ToPoint,  
    kMapPoint_ToNamedBlock,  
    kMapPoint_ToBlockFromData,  
    kMapPoint_ToTextFromData  
};
```

File

AcMapIEEnum.h

Parameters	Description
kMapPoint_Invalid	Invalid procedure.
kMapPoint_ToPoint	Import the point as a point.
kMapPoint_ToNamedBlock	Import the point as a named block.
kMapPoint_ToBlockFromData	Import the point as a block named by the specified column.
kMapPoint_ToTextFromData	Import the point as text.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIE Namespace](#)

AcMapIE:: StorageType Enumeration

[AcMapIE Namespace](#)

Enumerates the storage types of import and export formats. These enumerated values are returned by functions in the AcMapIEFormatclass and describe the data set type. A value can apply to only import, only export, or to both.

```
enum StorageType {
    kStorage_Invalid,
    kStorage_FileAllEntityTypes,
    kStorage_FileMultiSelect,
    kStorage_FileOneEntityType,
    kStorage_FolderWithPrefix,
    kStorage_FolderNoPrefix,
    kStorage_FolderOneEntityType,
    kStorage_FileMultiClassNone,
    kStorage_FileMultiClassLayer,
    kStorage_FileMultiClassObjectClass,
    kStorage_FileMultiClassObjectData,
    kStorage_FileMultiClassLinkTemplate,
    kStorage_FolderMultiClassNone,
    kStorage_FolderMultiClassLayer,
    kStorage_FolderMultiClassObjectClass,
    kStorage_FolderMultiClassObjectData,
    kStorage_FolderMultiClassLinkTemplate,
    kStorage_Database,
    kStorage_DatabaseMultiClassNone,
    kStorage_DatabaseMultiClassLayer,
    kStorage_DatabaseMultiClassObjectClass,
    kStorage_DatabaseMultiClassObjectData,
    kStorage_DatabaseMultiClassLinkTemplate
};
File
```

AcMapIEEnum.h

Parameters

kStorage\_Invalid

Description

Invalid storage type.

For import, the importer Init() expects a single file name. For export, the exporter Init() expects a single file

kStorage\_FileAllEntityTypes

kStorage_FileMultiSelect	name and the exporter can export all entity types.
kStorage_FileOneEntityType	For import, the importer Init() expects one or more file names. Does not apply to export.
kStorage_FolderWithPrefix	Does not apply to import. For export, the exporter Init() expects a single file name and the exporter can export only one entity type.
kStorage_FolderNoPrefix	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter can export all entity types.
kStorage_FolderOneEntityType	For import, the importer Init() expects a folder name. For export, the exporter Init() expects a folder name and the exporter can export all entity types.
kStorage_FileMultiClassNone	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter can export only one entity type.
kStorage_FileMultiClassLayer	Does not apply to import. For export, the exporter Init() expects a file name and the exporter will export a single class.
kStorage_FileMultiClassObjectClass	Does not apply to import. For export, the exporter Init() expects a file name and the exporter will export classes by layer.
kStorage_FileMultiClassObjectData	Does not apply to import. For export, the exporter Init() expects a file name and the exporter will export classes by object class.
	Does not apply to import. For export, the exporter Init() expects a file name and the exporter will export classes

kStorage_FileMultiClassLinkTemplate	by object data. Does not apply to import. For export, the exporter Init() expects a file name and the exporter will export classes by link template.
kStorage_FolderMultiClassNone	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter will export a single class.
kStorage_FolderMultiClassLayer	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter will export classes by layer.
kStorage_FolderMultiClassObjectClass	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter will export classes by object class.
kStorage_FolderMultiClassObjectData	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter will export classes by object data.
kStorage_FolderMultiClassLinkTemplate	Does not apply to import. For export, the exporter Init() expects a folder name and the exporter will export classes by link template.
kStorage_Database	Not supported.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Import-Export](#)

Classes

[Import-Export](#)

☐ Classes

[AcMapIEColumn](#)

Provides functions that handle imported columns (also called fields). For each imported column you define a mapping to a target column in AutoCAD Map. Two mappings are possible for each column: a feature class mapping and a data table mapping. (The feature class mapping is not set by default for any column.) Prior to using the mapping functions in this class, set the column's feature class and data table with

[AcMapIEInputLayer::SetFeatureClassName\(\)](#) and [AcMapIEInputLayer::SetDataMapping\(\)](#).

For more information, search for *importing* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in... [more](#)

[AcMapIEColumnIterator](#)

An iterator over a collection of `AcMapIEColumn` instances. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

Provides functions that handle the export process. For more information, search for *exporting* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

[AcMapIEExporter](#)

Provides callback functions for export reactor. To implement an export reactor, derive classes from these virtual base classes, overwriting the desired functions. Pass instances of the classes to the exporter (an instance of `AcMapIEExporter`) and the exporter will call back into your class.

## [AcMapIEExportReactor](#)

at various points in the export process. This behavior allows you to modify the export process at runtime or track the export process externally. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

## [AcMapIEExpressionTargetIterator](#)

An iterator over a collection of expression-target pairs. The case-insensitive target component of an expression-target pair is a key value and must be unique within the collection. This class (which is similar to the STL map class) differs from the [AcMapIENamespaceValueIterator](#) class in that this class has stricter naming rules for the target component of an expression-target pair (see the Add() or Set() function for naming rules). Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

## [AcMapIEFormat](#)

Provides functions that handle import and export file formats. For more information, search for *importing*, *import file formats*, *exporting*, and *export file formats* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

## [AcMapIEFormatIterator](#)

An iterator over a collection of AcMapIEFormat instances. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

## [AcMapIEImporter](#)

Provides functions that handle the import process. A default import operation brings in attribute data unless you change this behavior using functions in the AcMapIEInputLayer class. For more information, search for *importing* in

AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

Provides callback functions for import reactor. To implement an import reactor, derive class from these virtual base classes, overwriting the desired functions. Pass instances of the class to the importer (an instance of [AcMapIEImporter](#) and the importer will call back into your class at various points in the import process. This behavior allows you to modify the import process at runtime or track the import process externally. Do not explicitly delete any string returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

### [AcMapIEImportReactor](#)

Provides functions that handle input layers (also called themes).

After you call the [AcMapIEImporter](#) class's [LoadIPF\(\)](#), [InvokeDriverOptionsDialog\(\)](#), or [SetDriverOptions\(\)](#) function on the importer, a new [AcMapIEInputLayerIterator](#) object is created if the function's `bSchemaHasChanged` output parameter is true and any previously obtained input-layer iterator becomes invalid. Calling [AcMapIEImporter::Init\(\)](#) will also invalidate the [AcMapIEInputLayerIterator](#) because AutoCAD Map assumes you are opening a new dataset or changing format.

### [AcMapIEInputLayer](#)

If you set a table name and table type by using [SetDataMapping\(\)](#), default mappings are created depending on the existence of an object data table or a link template with that table name. valid table types,... [more](#)

An iterator over a collection of [AcMapIEInputLayer](#) instances. Do not explicitly delete any strings returned by any of the

### [AcMapIEInputLayerIterator](#)

## [AcMapIENameValueIterator](#)

functions or their output parameters in this class (but delete all other types of returned objects).  
An iterator over a collection of name-value pairs.  
The case-insensitive name component of a name-value pair is a key value and must be unique within the collection. This class (which is similar to the STL map class) differs from the [AcMapIEExpressionTargetIterator](#) class in that this class has more lenient naming rules for the name component of a name-value pair. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEColumn Class

[Classes](#)

Provides functions that handle imported columns (also called fields). For each imported column, you define a mapping to a target column in AutoCAD Map. Two mappings are possible for each column: a feature class mapping and a data mapping. (The feature class mapping is not set by default for any column.) Prior to using the mapping functions in this class, set the column's feature class and data table with [AcMapIEInputLayer::SetFeatureClassName\(\)](#) and [AcMapIEInputLayer::SetDataMapping\(\)](#).

For more information, search for *importing* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

**class** AcMapIEColumn;

File

AcMapIEColumn.h

☐ Methods

 <a href="#">~AcMapIEColumn</a>	Destroys an instance of this class.
<a href="#">ColumnClassMapping</a>	Returns the name of the feature class property that is mapped to this column.
<a href="#">ColumnDataMapping</a>	Returns the name of the column in the AutoCAD Map table that this incoming column is mapped to.
<a href="#">ColumnName</a>	Returns the (fixed) name of this incoming column.
<a href="#">SetColumnClassMapping</a>	Sets the feature class mapping of this incoming column.
<a href="#">SetColumnDataMapping</a>	Sets the name of the column in the AutoCAD Map table that this incoming column is mapped to. The mapped column name is set by default when you call <a href="#">AcMapIEInputLayer::SetDataMapping()</a> or reset every time you call <a href="#">SetColumnDataMapping()</a> with different data. The default behavior maps the incoming column name to an existing column in an

existing object data table, or an existing link or link template, where the case-insensitive names match. If no such target column exists then the incoming column is not mapped by default. When you create a new object data table as the default mapping, curly brace and... [more](#)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

AcMapIEColumn:: SetColumnClassMapping Method

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Sets the feature class mapping of this incoming column.

```
virtual AcMapIE::ErrCode SetColumnClassMapping(  
    const ACHAR* pszMappedPropertyName  
) = 0;
```

Parameters

Description

pszMappedPropertyName	Input name of the feature class property to map to this column. To reset the column to no mapping, use NULL or an empty string.
-----------------------	---

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_ClassNotSet if the class name was not first set with [AcMapIEInputLayer::SetFeatureClassName\(\)](#). Returns AcMapIE::ErrCode kErr\_BadFieldName if the column name does not exist in the specified feature class. Returns AcMapIE::ErrCode kErr\_Conflict if a conflict with a previously set column exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

AcMapIEColumn:: SetColumnDataMapping Method

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Sets the name of the column in the AutoCAD Map table that this incoming column is mapped to. The mapped column name is set by default when you call [AcMapIEInputLayer::SetDataMapping\(\)](#) or reset every time you call [SetColumnDataMapping\(\)](#) with different data. The default behavior maps the incoming column name to an existing column in an existing object data table, or an existing link or link template, where the case-insensitive names match. If no such target column exists then the incoming column is not mapped by default. When you create a new object data table as the default mapping, curly brace and dot characters are changed to underscores. Object data table column names must be 31 or fewer characters long. Object data table names must be 25 or fewer characters long.

```
virtual AcMapIE::ErrCode SetColumnDataMapping(  
    const ACHAR* pszMappedFieldName  
) = 0;
```

Parameters	Description
pszMappedFieldName	Input column name. To reset the column to no mapping, use NULL or an empty string.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_TableNotSet if the table name and table type was not first set with [AcMapIEInputLayer::SetDataMapping\(\)](#). Returns AcMapIE::ErrCode kErr\_BadFieldName if the column name does not exist in the specified table. Returns AcMapIE::ErrCode kErr\_Conflict if a conflict with a previously set feature class column exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEColumnIterator Class

[Classes](#)

An iterator over a collection of AcMapIEColumninstances. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEColumnIterator;
```

File

AcMapIEColumnIterator.h

☐ Methods



[~AcMapIEColumnIterator](#) Destroys an instance of this class.

[Done](#)

Determines whether the iterator has reached the end of the collection.

[Find](#)

Searches the collection for an element.

[Get](#)

Retrieves the current element in the iteration.

[Rewind](#)

Moves to the first element in the iteration.

[Step](#)

Advances to the next element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)

AcMapIEColumnIterator:: Find Method

[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Searches the collection for an element.

```
virtual AcMapIEColumn* Find(  
    const ACHAR * colName  
) = 0;
```

Parameters	Description
colName	Input name of the column to find.

Returns

Returns the named AcMapIEColumnif found; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEExporter Class

[Classes](#)

Provides functions that handle the export process. For more information, search for *exporting* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEExporter;
```

File

AcMapIEExporter.h

☐ Methods

[AddClassMapping](#)

Creates a per-class mapping when exporting multi-classes.

Adds a reactor to this exporter. You must derive a reactor from the virtual base class [AcMapIEExportReactor](#). Reactors are triggered through only the API; an end-user cannot trigger a reactor by using the AutoCAD Map user interface. Reactors are live until they are removed explicitly with `RemoveReactor()` or until AutoCAD Map exits. An exporter is a singleton; all callers add reactors to the same exporter instance. If an exporter has multiple reactors, they are called in the order added.

[AddReactor](#)

[ClosedPolylinesAsPolygons](#)

Determines whether closed polylines are exported as polygons.

[CountObjects](#)

Counts the number of objects to export in the current drawing.

[DiscretizationAngle](#)

Retrieves the discretization angle. The discretization angle setting is stored in the file *MapExport.ini*, and labeled `SegmentationDegrees`.

## [DriverOptions](#)

Retrieves the current driver options. Driver options can be set in the file *MapExport.ini*, by calling [InvokeDriverOptionsDialog\(\)](#), or by loading a profile. Use this function to first retrieve the current driver options before adding driver options with [SetDriverOptions\(\)](#).

## [Export](#)

Performs the export.

## [ExportAll](#)

Determines whether a selection set is ignored during export.

## [ExportDataMappings](#)

Retrieves the source-column and output-column data mappings for export. The output collection will be empty if you have not first loaded an export profile file with [LoadEPF\(\)](#).

## [ExportFromPolygonTopology](#)

Retrieves the settings for the polygon topology to export.

## [FeatureClassFilter](#)

Retrieves the feature classes as a list of comma-separated patterns.

## [FormatName](#)

Returns the name of the format set during initialization. The format name is set with [Init\(\)](#).

## [Init](#)

Initializes an instance of this class.

## [InvokeDriverOptionsDialog](#)

Invokes the driver-specific Driver Options dialog box. Not all drivers have dialog boxes, so this function may finish immediately, with the returned error code indicating the status of the dialog box. Any setting changes that an AutoCAD Map user makes in this dialog box are reflected in subsequent calls to [DriverOptions\(\)](#). You can set driver options programmatically by using [SetDriverOptions\(\)](#). Any driver options in the file *MapExport.ini* are read and available in the set of driver options returned by [DriverOptions\(\)](#).

## [LayerFilter](#)

Retrieves the layers as a list of comma-separated patterns.

<a href="#"><u>LayerLevelMapping</u></a>	Retrieves the settings for mapping layers in the drawing to DGN levels.
<a href="#"><u>LoadEPF</u></a>	Loads an export profile (.
<a href="#"><u>RemoveReactor</u></a>	Removes a reactor from this exporter.
<a href="#"><u>ResetDataTypeForProperty</u></a>	Sets the Data Type for new or existing properties.
<a href="#"><u>SaveEPF</u></a>	Saves an export profile (.
<a href="#"><u>SelectionSet</u></a>	Retrieves the selection set.
<a href="#"><u>SetClassMappingType</u></a>	Sets the Create-By type when using Per-Class Mapping for FDO targets.
<a href="#"><u>SetClosedPolylinesAsPolygons</u></a>	Changes the setting for exporting closed polylines as polygons.
<a href="#"><u>SetDiscretizationAngle</u></a>	Sets the discretization angle. The discretization angle setting is stored in the file <i>MapExport.ini</i> , and labeled SegmentationDegrees.
<a href="#"><u>SetDriverOptions</u></a>	Sets the driver options. Typically, you should retrieve the current driver options from DriverOptions()and make changes as needed by using this function.
<a href="#"><u>SetExportAll</u></a>	Changes the setting for using a selection set during export.
<a href="#"><u>SetExportDataMappings</u></a>	Sets the source-column and output-column data mappings for export.
<a href="#"><u>SetExportFromPolygonTopology</u></a>	Changes the settings for the polygon topology to export.
<a href="#"><u>SetFeatureClassFilter</u></a>	Sets the feature classes.
<a href="#"><u>SetGeometryTypeForClass</u></a>	Sets the Geometry Type for new target class.
<a href="#"><u>SetIsUniqueKeyProperty</u></a>	Sets a property to be a Unique/Primary Key.
<a href="#"><u>SetLayerFilter</u></a>	Sets the layer patterns.
<a href="#"><u>SetLayerLevelMapping</u></a>	Changes the settings for mapping layers in the drawing to DGN levels.
<a href="#"><u>SetSelectionSet</u></a>	Sets the ADS (AutoCAD Development System) selection set.
	Sets the storage options. Format-specific

### [SetStorageOpts](#)

storage options are stored in the file *MapExport.ini*. This function attempts to make sense of its input values even if those values are inconsistent with the storage type. Format support changes dynamically according to the installed drivers so, because it is impossible to know in advance which drivers are installed, this function cannot always determine which combinations of input values make sense for a particular format; this function's behavior is undefined when using invalid input values. Only a few cases exist where it makes sense to change the storage type value from that returned... [more](#)

### [SetTargetCoordSys](#)

Sets the target coordinate system for the exported data.

### [SetUseUniqueKeyField](#)

Changes the settings for creating a unique value for each exported object.

### [StorageOpts](#)

Retrieves the storage options. Format-specific storage options are stored in the file *MapExport.ini*.

### [TargetCoordSys](#)

Retrieves the target coordinate system for the exported data.

### [UseUniqueKeyField](#)

Determines whether a unique value is created for each exported object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: AddClassMapping Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Creates a per-class mapping when exporting multi-classes.

```
virtual AcMapIE::ErrCode AddClassMapping(  
    const AcMapIEExportClassMapping*& mapping  
) = 0;
```

Parameters	Description
mapping	The Class Mapping
Returns	

kErr\_OK if successful

kErr\_CannotModifySchema if the datastore does not have that capability

kErr\_Conflict if global mappings have also been set

kErr\_InvalidMapping if the source has already been mapped

kErr\_InvalidDataStore if the target datastore is not supported (FDO)

Remarks

The source type for each mapping is based on the StorageOpts setting: Layers, Object Data, Object Classes, or Link Templates. This mapping can only be used with FDO data stores as the target. If this mapping is set, then any global mappings created by [SetLayerLevelMapping\(\)](#) and [SetExportDataMappings\(\)](#) are not used. Any new Schema, Class, or Property names which appear in the mapping, but not in an existing datastore will be appended - if the datastore has an append capability.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::ExportFromPolygonTopology Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the settings for the polygon topology to export.

```
virtual AcMapIE::ErrCode ExportFromPolygonTopology(  
    bool& bGroupComplex,  
    const ACHAR*& pszTopologyName  
) const = 0;
```

Parameters	Description
bGroupComplex	Output true if complex polygons are grouped when exporting this topology; otherwise, false.
pszTopologyName	Output name of the topology.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: Init Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Initializes an instance of this class.

```
virtual AcMapIE::ErrCode Init(  
    const ACHAR* pszFormatName,  
    const ACHAR* pszFileName  
) = 0;
```

Parameters	Description
pszFormatName	Input name of an AcMapIEFormatobject.
pszFileName	Input fully qualified file or directory name, depending on the format.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if the format is invalid. Returns AcMapIE::ErrCode kErr\_FileError if the file or directory name is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

Call this function before any other export function (or that function will return AcMapIE::ErrCode kErr\_NotInitialized).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::LoadEPF Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Loads an export profile (.

```
virtual AcMapIE::ErrCode LoadEPF(  
    const ACHAR* pszFileName  
) = 0;
```

Parameters	Description
pszFileName	Input fully qualified file name. The filename extension .epf is added automatically if it is omitted. Any driver options saved in the .epf will be loaded if they apply to the same format that the exporter was initialized for.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_FileError if the file could not be loaded. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::ResetDataTypeForProperty Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the Data Type for new or existing properties.

```
virtual AcMapIE::ErrCode ResetDataTypeForProperty(  
    const ACHAR* schemaName,  
    const ACHAR* className,  
    const ACHAR* propertyName,  
    AcMapIE::DataType dataType  
) = 0;
```

Parameters	Description
schemaName	Name of the target schema
className	Name of the target class
propertyName	Name of the target property
dataType	Data Type of the property

## Returns

kErr\_OK if successful

kErr\_ClassExists if the property is not a newly appended one

kErr\_InvalidMapping if the property is not mapped to

kErr\_InvalidDataStore if the target datastore is not supported (FDO)

## Remarks

Normally the data type is automatically determined based on the source data type. This setting allows the type to be reset to another type so that expressions can be used to convert the source type. For example, a source type of .Color can be mapped to Reals as pipe diameters. This cannot be used on already existing properties.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SaveEPF Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Saves an export profile (.

```
virtual AcMapIE::ErrCode SaveEPF(  
    const ACHAR* pszFileName  
) = 0;
```

Parameters	Description
pszFileName	Input fully qualified file name. The filename extension .epf is added automatically if it is omitted.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_FileError if the file could not be saved. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)  
AcMapIEExporter:: SetClassMappingType Method  
[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the Create-By type when using Per-Class Mapping for FDO targets.

```
virtual AcMapIE::ErrCode SetClassMappingType(  
    AcMapIE::ExportClassMapping mappingType  
) = 0;
```

Parameters	Description
mappingType	Type of source objects the class mapping is using, such as Layer, Object Data Tables.

## Remarks

Do not reset this unless using Per-Class Mapping. This must be called before calling AddClassMapping().

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::SetExportFromPolygonTopology Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Changes the settings for the polygon topology to export.

```
virtual AcMapIE::ErrCode SetExportFromPolygonTopology(  
    bool bGroupComplex,  
    const ACHAR* pszTopologyName  
) = 0;
```

Parameters	Description
bGroupComplex	Input true to group complex polygons when exporting; otherwise, false.
pszTopologyName	Input name of the topology.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

This function does not check if the specified topology exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetFeatureClassFilter Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the feature classes.

```
virtual AcMapIE::ErrCode SetFeatureClassFilter(  
    const ACHAR* pszClassList  
) = 0;
```

Parameters

Description

pszClassList

Input feature class list, as a list of comma-separated patterns.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::SetGeometryTypeForClass Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the Geometry Type for new target class.

```
virtual AcMapIE::ErrCode SetGeometryTypeForClass(  
    const ACHAR* schemaName,  
    const ACHAR* className,  
    AcMapIE::GeometryType geomType  
) = 0;
```

Parameters	Description
schemaName	Name of the target schema
className	Name of the target class
geomType	Type of Geometry to set. Currently kGeom_Text is not supported

Returns

kErr\_OK if successful

kErr\_ClassExists if the class is not a newly appended one

kErr\_InvalidMapping if the class is not mapped to

kErr\_BadParams if the unsupported kGeom\_Text is passed in

kErr\_InvalidDataStore if the target datastore is not supported (FDO)

Remarks

This cannot be called on existing classes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetIsUniqueKeyProperty Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets a property to be a Unique/Primary Key.

```
virtual AcMapIE::ErrCode SetIsUniqueKeyProperty(  
    const ACHAR* schemaName,  
    const ACHAR* className,  
    const ACHAR* propertyName  
) = 0;
```

Parameters	Description
schemaName	Name of the target schema
className	Name of the target class
propertyName	Name of the target property

## Returns

kErr\_OK if successful

kErr\_ClassExists if the property is not a newly appended one

kErr\_InvalidDataStore if the target datastore is not supported (FDO)

## Remarks

This can only be called for new properties. The property does not need to have a mapping - and should not, if it is an autogenerated key field. In that case, the new property will be added as an autogenerated Unique Key field.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter:: SetLayerFilter Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the layer patterns.

```
virtual AcMapIE::ErrCode SetLayerFilter(  
    const ACHAR* pszLayerList  
) = 0;
```

Parameters

Description

pszLayerList

Input layer list, as a list of comma-separated patterns.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetStorageOpts Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the storage options. Format-specific storage options are stored in the file *MapExport.ini*. This function attempts to make sense of its input values even if those values are inconsistent with the storage type. Format support changes dynamically according to the installed drivers so, because it is impossible to know in advance which drivers are installed, this function cannot always determine which combinations of input values make sense for a particular format; this function's behavior is undefined when using invalid input values. Only a few cases exist where it makes sense to change the storage type value from that returned by `StorageOpts()`. The following `StorageType` storage types are valid for export: [AcMapIE::StorageType](#) `kStorage_FileAllEntityType` [AcMapIE::StorageType](#) `kStorage_FileOneEntityType` [AcMapIE::StorageType](#) `kStorage_FolderWithPrefix` [AcMapIE::StorageType](#) `kStorage_FolderNoPrefix` [AcMapIE::StorageType](#) `kStorage_FolderOneEntityType` If the storage type is `kStorage_FolderWithPrefix`, then set a prefix. Use `kStorage_FolderWithPrefix` to set up a Shape format as a folder type. The output will consist of up to four shape files, named `<prefix>_point.shp`, `<prefix>_line.shp`, `<prefix>_polygon.shp`, and `<prefix>_text.shp`. The other possible storage type for the Shape format is `kStorage_FileOneEntityType`. It is possible to change the storage type between these two values for the Shape format. Among the formats that AutoCAD Map supports, Shape is the only one that requires a geometry type and prefix. For the ARCINFO format, corresponding to Coverage, the storage type can be either `kStorage_FolderNoPrefix` or `kStorage_FolderOneEntityType`. Other formats have only one valid storage type, which is the type returned by `StorageOpts()`. If the storage type is `kStorage_FileOneEntityType` or `kStorage_FolderOneEntityType` then the default geometry type is `kGeom_Point`. Other storage types will ignore the geometry type setting. To export different types of entities, set `geomType` to `kGeom_Point`, `kGeom_Line`, `kGeom_Polygon`, or `kGeom_Text`. AutoCAD Map will filter out entities of incorrect types before exporting.

```
virtual AcMapIE::ErrCode SetStorageOpts(  
    AcMapIE::StorageType storageType,  
    AcMapIE::GeometryType geomType,  
    const ACHAR* pszPrefix
```

) = 0;

Parameters	Description
storageType	Input StorageTypestorage type.
geomType	Input GeometryTypegeometry type.
pszPrefix	Input prefix.
Returns	

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetTargetCoordSys Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the target coordinate system for the exported data.

```
virtual AcMapIE::ErrCode SetTargetCoordSys(  
    const ACHAR* pszCoordName  
) = 0;
```

Parameters	Description
pszCoordName	Input name of the target coordinate system to set.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if the coordinate system name is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetUseUniqueKeyField Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Changes the settings for creating a unique value for each exported object.

```
virtual AcMapIE::ErrCode SetUseUniqueKeyField(  
    const ACHAR* pszKeyFieldName  
) = 0;
```

Parameters	Description
pszKeyFieldName	Input unique-key column name to set. The column name must be 255 or fewer characters long and can include the following symbols: a-z A-Z 0-9 . { } - _ . (Including accented and multibyte characters.)

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadFieldName if the column name is too long or contains invalid characters. Returns AcMapIE::ErrCode kErr\_ConflictWithTarget if the column name already exists as a mapped target column name. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter::StorageOpts Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the storage options. Format-specific storage options are stored in the file *MapExport.ini*.

```
virtual AcMapIE::ErrCode StorageOpts(  
    AcMapIE::StorageType& storageType,  
    AcMapIE::GeometryType& geomType,  
    const ACHAR*& pszPrefix  
) const = 0;
```

Parameters	Description
storageType	Output StorageType storage type.
geomType	Output GeometryType geometry type.
pszPrefix	Output prefix.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::UseUniqueKeyField Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Determines whether a unique value is created for each exported object.

```
virtual AcMapIE::ErrCode UseUniqueKeyField(  
    const ACHAR*& pszKeyFieldName  
) const = 0;
```

Parameters	Description
pszKeyFieldName	Output name of the unique key column.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

The default key column in the AutoCAD Map user interface is AcMapKey, but the key column has no default value in this API.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEExportReactor Class

[Classes](#)

Provides callback functions for export reactors. To implement an export reactor, derive classes from these virtual base classes, overwriting the desired functions. Pass instances of the classes to the exporter (an instance of AcMapIEExporter) and the exporter will call back into your classes at various points in the export process. This behavior allows you to modify the export process at runtime or track the export process externally. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

**class** AcMapIEExportReactor;

File

AcMapIEReactor.h

☐ Methods



[~AcMapIEExportReactor](#) Destroys an instance of this class.



[AcMapIEExportReactor](#) Constructs an instance of this class.

[RecordError](#) Invoked if an error occurs during export.

[RecordExported](#) Invoked after an entity is exported.

[RecordReadyForExport](#) Invoked before an entity is exported.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExportReactor Class](#), [AcMapIEExportReactor Class](#)

[AcMapIEExportReactor::RecordError Method](#)

[AcMapIEExportReactor Class](#) | [AcMapIEExportReactor Class](#)

Invoked if an error occurs during export.

```
virtual void RecordError(  
    const ACHAR* error,  
    AcDbEntity* pEnt  
);
```

Parameters	Description
error	Input error description.
pEnt	Input entity.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEExpressionTargetIterator Class

[Classes](#)

An iterator over a collection of expression-target pairs. The case-insensitive target component of an expression-target pair is a key value and must be unique within the collection. This class (which is similar to the STL map class) differs from the [AcMapIENameValueIterator](#) class in that this class has stricter naming rules for the target component of an expression-target pair (see the Add() or Set() function for naming rules). Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEExpressionTargetIterator;
```

File

AcMapIEExpressionTargetIterator.h

☐ Methods



[~AcMapIEExpressionTargetIterator](#)

Destroys an instance of this class.

[Add](#)

Inserts a new element into the collection after the current element and advances the iterator to the new element.

[Clear](#)

Removes all elements from the collection.

[Done](#)

Determines whether the iterator has reached the end of the collection.

[Find](#)

Searches the collection for an element.

[Get](#)

Retrieves the current element in the iteration.

[Remove](#)

Removes an element from the collection.

[Rewind](#)

Moves to the first element in the iteration.

[Set](#)

Sets the expression of an expression-target pair in the collection.

[Step](#)

Advances to the next element in the

iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Add Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Inserts a new element into the collection after the current element and advances the iterator to the new element.

```
virtual AcMapIE::ErrCode Add(  
    const ACHAR* pszExpression,  
    const ACHAR* pszTarget  
) = 0;
```

Parameters	Description
pszExpression	Input expression of the expression-target pair to add.
pszTarget	Input target of the expression-target pair to add.
Returns	

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadParams if either input string is NULL or empty. Returns AcMapIE::ErrCode kErr\_BadFieldName if the input target is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the case-insensitive input target already exists in the collection.

## Remarks

The name of the target must be 255 or fewer characters long and can include the following symbols: a-z A-Z 0-9 . { } - \_ . The target to which you are exporting may have stricter naming rules, however, so it is safest to restrict target names to alphanumeric, multi-byte, and underscore characters; in particular, avoid these characters: . { }.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Find Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Searches the collection for an element.

```
virtual bool Find(  
    const ACHAR*& pszExpression,  
    const ACHAR * pszTarget  
) = 0;
```

Parameters	Description
pszExpression	Output expression corresponding to the input target if the expression-target pair was found. An empty string indicates that the element was not found.
pszTarget	Input case-insensitive target of the expression-target pair to find.

Returns

Returns true if the element was found; otherwise, returns false. False indicates an invalid output expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Get Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Retrieves the current element in the iteration.

```
virtual bool Get(  
    const ACHAR*& pszExpression,  
    const ACHAR*& pszTarget  
) = 0;
```

Parameters

Description

pszExpression

Output expression of the current expression-target pair.

pszTarget

Output target of the current expression-target pair.

Returns

Returns true if the current element is valid, or false if the iteration has no more elements to retrieve.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Remove Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Removes an element from the collection.

```
virtual bool Remove(  
    const ACHAR* pszTarget  
) = 0;
```

Parameters

Description

pszTarget

Input case-insensitive target of the expression-target pair to remove.

Returns

Returns true if the element was removed successfully; otherwise, returns false.

Remarks

If the iterator is pointing to the removed element, it advances to the next element if possible.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Set Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Sets the expression of an expression-target pair in the collection.

```
virtual AcMapIE::ErrCode Set(  
    const ACHAR* pszExpression,  
    const ACHAR* pszTarget  
) = 0;
```

Parameters	Description
pszExpression	Input new expression for the expression-target pair to set.
pszTarget	Input case-insensitive target of the expression-target pair to set.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadParams if either input string is NULL or empty. Returns AcMapIE::ErrCode kErr\_BadFieldName if the input target name is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the case-insensitive input target does not already exist in the collection.

## Remarks

The name of the target must be 255 or fewer characters long and can include the following symbols: a-z A-Z 0-9 . { } - \_ . The target to which you are exporting may have stricter naming rules, however, so it is safest to restrict target names to alphanumeric and underscore characters; in particular, avoid these characters: . { }.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEFormat Class

[Classes](#)

Provides functions that handle import and export file formats. For more information, search for *importing*, *import file formats*, *exporting*, and *export file formats* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEFormat;
```

File

AcMapIEFormat.h

☐ Methods

◆ [~AcMapIEFormat](#)

Destroys an instance of this class.

[Description](#)

Retrieves a brief description of this format.

[Extension](#)

Retrieves the filename extension that AutoCAD Map expects to be used for this format.

[FormatName](#)

Returns the unique name which AutoCAD Map uses to refer to this format. This format name is not translated and is used in the AutoCAD Map configuration files *MapImport.ini*, *MapExport.ini*, and *MapForeignFileProperties.ini*.

[HasDriverOptionsDialog](#)

Determines whether a Driver Options dialog box exists for this format.

[StorageType](#)

Returns the storage type of this format.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEFormatIterator Class

[Classes](#)

An iterator over a collection of AcMapIEFormatinstances. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEFormatIterator;
```

File

AcMapIEFormatIterator.h

☐ Methods

[Done](#)

Determines whether the iterator has reached the end of the collection.

[Find](#)

Searches the collection for an element.

[Get](#)

Retrieves the current element in the iteration.

[Rewind](#)

Moves to the first element in the iteration.

[Step](#)

Advances to the next element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormatIterator Class](#), [AcMapIEFormatIterator Class](#)

[AcMapIEFormatIterator:: Find Method](#)

[AcMapIEFormatIterator Class](#) | [AcMapIEFormatIterator Class](#)

Searches the collection for an element.

```
virtual AcMapIEFormat* Find(  
    const ACHAR * pszFormatName  
) = 0;
```

Parameters	Description
pszFormatName	Input name of the format to find.

Returns

Returns the named AcMapIEFormatif found; otherwise, returns null. The caller is responsible for deleting this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEImporter Class

[Classes](#)

Provides functions that handle the import process. A default import operation brings in no attribute data unless you change this behavior by using functions in the AcMapIEInputLayerclass. For more information, search for *importing* in AutoCAD Map Help. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEImporter;  
File
```

AcMapIEImporter.h

☐ Methods

[AddReactor](#)

Adds a reactor to this importer. You must derive a reactor from the virtual base class [AcMapIEImportReactor](#). Reactors are triggered through only the API; an end-user cannot trigger a reactor by using the AutoCAD Map user interface. Reactors are live until they are removed explicitly with `RemoveReactor()` or until AutoCAD Map exits. An importer is a singleton; all callers add reactors to the same importer instance. If an importer has multiple reactors, they are called in the order added.

[AuditClassifiedAfterImport](#)

Determines whether classified properties are audited after import.

Retrieves the current driver options. Driver options can be set in the file *MapImport.ini*, by calling [InvokeDriverOptionsDialog\(\)](#), or by

## [DriverOptions](#)

loading a profile. Use this function to first retrieve the current driver options before adding driver options with `SetDriverOptions()`.

## [Import](#)

Performs the import.

## [ImportPolygonsAsClosedPolylines](#)

Determines whether polygons are imported as closed polylines or as polygons.

## [Init](#)

Initializes an instance of this class. Call this function before any other import function (or that function will return `AcMapIE::ErrCode kErr_NotInitialized`). You must call `Init()` again after every call to `Import()` (with one exception: you can call `RemoveReactor()` after `Import()` without first calling `Init()`).

## [InputLayerIterator](#)

Retrieves an iterator over the input layers. An input layer refers to a single MIF/MID or Shape file, a level or geometry type in a DGN file, or a geometry type in a Coverage or E00 file. A layer consists of a layer name and a schema for attribute data. You are responsible for determining how this data is imported; for more information, see the `AcMapIEInputLayer` and `AcMapIEColumn` classes.

## [InvokeDriverOptionsDialog](#)

Invokes the driver-specific Driver Options dialog box. Not all drivers have dialog boxes, so this function may finish immediately, with the returned error code indicating the status of the dialog box. Any setting changes that an AutoCAD Map user makes in this dialog box are reflected in subsequent calls to `DriverOptions()`. You can set driver

options programmatically by using `SetDriverOptions()`. Any driver options in the file *MapImport.ini* are read and available in the set of driver options returned by `DriverOptions()`.

[LoadIPF](#)

Loads an import profile (.).

[LocationWindowAndOptions](#)

Determines which location window is used to filter incoming entities.

[RemoveReactor](#)

Removes a reactor from this importer.

[SaveIPF](#)

Saves an import profile (.).

[SetAuditClassifiedAfterImport](#)

Enables or disables classified-property audits after import.

[SetDriverOptions](#)

Sets the driver options. Typically, you should retrieve the current driver options from `DriverOptions()` and make changes as needed by using this function.

[SetImportPolygonsAsClosedPolylines](#)

Sets the manner in which polygons are imported: as closed polylines or as polygons.

[SetLocationWindowAndOptions](#)

Sets the location window used to filter incoming entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

[AcMapIEImporter:: Init Method](#)

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Initializes an instance of this class. Call this function before any other import function (or that function will return `AcMapIE::ErrCode kErr_NotInitialized`). You must call `Init()` again after every call to `Import()` (with one exception: you can call `RemoveReactor()` after `Import()` without first calling `Init()`).

```
virtual AcMapIE::ErrCode Init(  
    const ACHAR* pszFormatName,  
    const ACHAR* pszFileName  
) = 0;
```

Parameters	Description
<code>pszFormatName</code>	Input name of an <code>AcMapIEFormat</code> object.
<code>pszFileName</code>	Input fully qualified file or directory name, depending on the format.

## Returns

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_Invalid` if the format is invalid. Returns `AcMapIE::ErrCode kErr_FileError` if the file or directory name is invalid. Returns `AcMapIE::ErrCode kErr_BadParams` if either input name is `NULL` or empty. Returns `AcMapIE::ErrCode kErr_Fail` if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter::LoadIPF Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Loads an import profile (.

```
virtual AcMapIE::ErrCode LoadIPF(  
    const ACHAR* pszFileName,  
    bool& bSchemaHasChanged  
) = 0;
```

Parameters	Description
pszFileName	Input fully qualified file name. The filename extension .ipf is added automatically if it is omitted.
bSchemaHasChanged	Output true if the schema has changed; otherwise, false. If this value is true, then any input-layer iterator that you obtained previously becomes invalid.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_FileError if the file could not be loaded. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: SaveIPF Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Saves an import profile (.ipf)

```
virtual AcMapIE::ErrCode SaveIPF(  
    const ACHAR* pszFileName  
) = 0;
```

Parameters

Description

pszFileName

Input fully qualified file name. The filename extension .ipf is added automatically if it is omitted.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_FileError if the file could not be saved. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEImportReactor Class

[Classes](#)

Provides callback functions for import reactors. To implement an import reactor, derive classes from these virtual base classes, overwriting the desired functions. Pass instances of the classes to the importer (an instance of AcMapIEImporter) and the importer will call back into your classes at various points in the import process. This behavior allows you to modify the import process at runtime or track the import process externally. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

**class** AcMapIEImportReactor;

File

AcMapIEReactor.h

☐ Methods



[~AcMapIEImportReactor](#) Destroys an instance of this class.



[AcMapIEImportReactor](#) Constructs an instance of this class.

[RecordError](#) Invoked if an error occurs during import.

[RecordImported](#) Invoked after an entity is imported.

[RecordReadyForImport](#) Invoked before an entity is imported.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImportReactor Class](#), [AcMapIEImportReactor Class](#)

[AcMapIEImportReactor::RecordError Method](#)

[AcMapIEImportReactor Class](#) | [AcMapIEImportReactor Class](#)

Invoked if an error occurs during import.

```
virtual void RecordError(  
    const ACHAR* pError  
);
```

Parameters	Description
pError	Input error description.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEInputLayer Class

[Classes](#)

Provides functions that handle input layers (also called themes).

After you call the AcMapIEImporter class's LoadIPF(), InvokeDriverOptionsDialog(), or SetDriverOptions() function on the importer, a new [AcMapIEInputLayerIterator](#) object is created if the function's bSchemaHasChanged output parameter is true and any previously obtained input-layer iterator becomes invalid. Calling [AcMapIEImporter::Init\(\)](#) will also invalidate the [AcMapIEInputLayerIterator](#) because AutoCAD Map assumes you are opening a new dataset or changing format.

If you set a table name and table type by using SetDataMapping(), default mappings are created depending on the existence of an object data table or a link template with that table name. The valid table types, set by the input [AcMapIE::ImportDataMapping](#) tableType parameter, are kNewObjectDataOnly, kExistingObjectDataOnly, kLinkTemplate, and kLinkOnly.

If the table type is kNewObjectDataOnly and the table name is an existing object data table, then SetDataMapping() generates an error and makes no changes. If no object data table with the specified name exists, then all incoming columns are mapped by default to valid object data column names. During import, when the first entity arrives in this input layer, the table is created. If no entities exist for this input layer, no object data table is created.

If the table type is kExistingObjectDataOnly and the table name is an existing object data table, then SetDataMapping() generates an error and makes no changes. If the object data table exists, then incoming columns are mapped to object data columns that have the same (case-insensitive) names.

If the table type is kLinkTemplate or kLinkOnly and the table name is not an existing link template, then SetDataMapping() generates an error and makes no changes. If the link template exists, then incoming columns are mapped to link template or link columns that have the same (case-insensitive) names. All link templates contain link columns which are the equivalent of unique key columns. If not all link columns are mapped to incoming columns, the mapping is invalid

and no data will be imported (and a warning is generated when setting the data mapping). It is your responsibility to ensure that all link columns are mapped. The database that you are importing data into may itself have constraints. If a key-column uniqueness constraint is violated, data may disappear silently.

```
class AcMapIEInputLayer;  
File
```

AcMapIEInputLayer.h

#### ☐ Methods

<a href="#">◆ ~AcMapIEInputLayer</a>	Destroys an instance of this class.
<a href="#">ColumnIterator</a>	Retrieves an iterator over a collection of AcMapIEColumninstances.
<a href="#">DataMapping</a>	Retrieves the table type and table name of the data mapping.
<a href="#">FeatureClassName</a>	Retrieves the name of the feature class of this input layer.
<a href="#">ImportFromInputLayerOn</a>	Determines whether this import layer is on (and will be imported).
<a href="#">LayerName</a>	Retrieves the name of the target layer to which entities are imported.
<a href="#">Name</a>	Retrieves the name of this input layer.
<a href="#">OriginalCoordSys</a>	Retrieves AutoCAD Map's best guess for the coordinate system that the data to be imported uses.
<a href="#">PointToBlockMapping</a>	Determines how points are imported.
<a href="#">SetDataMapping</a>	Sets the table type and table name of the data mapping.
<a href="#">SetFeatureClassName</a>	Sets the name of the feature class of this input layer.
<a href="#">SetImportFromInputLayerOn</a>	Turns this input layer on (to be imported) or off (not to be imported).
<a href="#">SetLayerName</a>	Sets the name of the target layer to which entities are imported.
<a href="#">SetPointToBlockMapping</a>	Sets how points are imported. Sets the coordinate system that incoming data is

[SetTargetCoordSys](#)

transformed from.

[SetUseForBlockAttributes](#)

Sets whether block attribute values are brought in from incoming attribute data.

[SetUseUniqueKeyField](#)

Sets whether a unique key value is created for each incoming entity. This setting is ignored unless an existing or new object data table already has been set with `SetDataMapping()`. The key column has type integer. In the AutoCAD Map user interface, this setting corresponds to the Add Unique Key Field check box in the Attribute Data dialog box.

[TargetCoordSys](#)

Retrieves the coordinate system that incoming data is transformed from.

[UseForBlockAttributes](#)

Determines whether block attribute values are brought in from incoming attribute data.

[UseUniqueKeyField](#)

Determines whether a unique value is created for each incoming entity. This setting is ignored unless an existing or new object data table already has been set with `SetDataMapping()`. The key column has type integer. In the AutoCAD Map user interface, this setting corresponds to the Add Unique Key Field check box in the Attribute Data dialog box.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

AcMapIEInputLayer::DataMapping Method

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves the table type and table name of the data mapping.

```
virtual AcMapIE::ErrCode DataMapping(  
    AcMapIE::ImportDataMapping& tableType,  
    const ACHAR*& pszTableName  
) const = 0;
```

Parameters	Description
tableType	Output ImportDataMappingtype of table used. For a detailed description of this parameter, see the overview of this class.
pszTableName	Output table name (object data or link template, depending on tableType).

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Remarks

In the AutoCAD Map user interface, this setting corresponds to the Data column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: FeatureClassName Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves the name of the feature class of this input layer.

```
virtual AcMapIE::ErrCode FeatureClassName(  
    const ACHAR*& pszName  
) const = 0;
```

Parameters	Description
pszName	Output name of the feature class.
Returns	

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

Use the column class to map individual columns. In the AutoCAD Map user interface, this setting corresponds to the Feature Class column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

AcMapIEInputLayer::LayerName Method

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves the name of the target layer to which entities are imported.

```
virtual AcMapIE::ErrCode LayerName(  
    AcMapIE::LayerNameType& layerNameType,  
    const ACHAR*& pszName  
) const = 0;
```

Parameters	Description
layerNameType	Output LayerNameTypevalue indicating how the layer name is used in the importer. The value kLayerName_Direct means that layer name specified by pszName is used. The value kLayerName_Indirect means that entities from this input layer are placed on the layer specified by the column named pszName.
pszName	Output name of the layer or column, depending on layerNameType.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Drawing Layer column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: PointToBlockMapping Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Determines how points are imported.

```
virtual AcMapIE::ErrCode PointToBlockMapping(  
    AcMapIE::PointMappingType& howToMap,  
    const ACHAR*& pszName  
) const = 0;
```

Parameters	Description
howToMap	Output enumerated value that indicates how points are imported.
pszName	Output column name for the text string or block name, depending on howToMap.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Points column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

AcMapIEInputLayer::SetDataMapping Method

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets the table type and table name of the data mapping.

```
virtual AcMapIE::ErrCode SetDataMapping(  
    const AcMapIE::ImportDataMapping tableType,  
    const ACHAR* pszTableName  
) = 0;
```

Parameters	Description
tableType	Input ImportDataMappingtype of table to use. For a detailed description of this parameter, see the overview of this class.
pszTableName	Input table name. This table name can be an existing object data table, a new object data table, or an existing link template (depending on tableType).

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if tableType is kNewObjectDataOnly and pszTableName is an existing table; if tableType is kExistingObjectDataOnly and pszTableName is not an existing table; or if tableType is kLinkTemplate and pszTableName is not an existing link template. Returns AcMapIE::ErrCode kErr\_BadTableName if the new data table name is invalid. Allowed characters are alphanumeric, multibyte, and - \_ and maximum length is 25 characters. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: SetFeatureClassName Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets the name of the feature class of this input layer.

```
virtual AcMapIE::ErrCode SetFeatureClassName(  
    const ACHAR* pszName  
) = 0;
```

Parameters	Description
pszName	Input name of the feature class to set.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if the feature class name is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

Use the column class to map individual columns. In the AutoCAD Map user interface, this setting corresponds to the Feature Class column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer::SetLayerName Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets the name of the target layer to which entities are imported.

```
virtual AcMapIE::ErrCode SetLayerName(  
    AcMapIE::LayerNameType layerNameType,  
    const ACHAR* pszName  
) = 0;
```

Parameters	Description
layerNameType	Input LayerNameTypevalue specifying how to use the layer name in the importer. Set this parameter to kLayerName_Direct to use the layer name specified by pszName. Set this parameter to kLayerName_Indirect to place the entities from this input layer on the layer specified by the column named by pszName.
pszName	Input name of a layer or column, depending on layerNameType.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadFieldName if layerNameType is kLayerName\_Indirect and the column pszName does not exist. Returns AcMapIE::ErrCode kErr\_BadParams if layerNameType is kLayerName\_Direct and pszName is empty. Returns AcMapIE::ErrCode kErr\_Invalid if layerNameType is kLayerName\_Direct and pszName is an invalid layer name. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Drawing Layer column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer::SetPointToBlockMapping Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets how points are imported.

```
virtual AcMapIE::ErrCode SetPointToBlockMapping(  
    AcMapIE::PointMappingType howToMap,  
    const ACHAR* pszName  
) = 0;
```

Parameters	Description
howToMap	Input enumerated value that determines how points are imported.
pszName	Input column name for the text string or block name, depending on howToMap.

## Returns

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_NoBlocksInDrawing` if `howToMap` sets a block-mapping type in a drawing with no blocks. Returns `AcMapIE::ErrCode kErr_BadFieldName` if `howToMap` is `kMapPoint_ToBlockFromData` or `kMapPoint_ToTextFromData` and the column name is invalid. Returns `AcMapIE::ErrCode kErr_Fail` if the process failed for some other reason.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Points column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: SetTargetCoordSys Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets the coordinate system that incoming data is transformed from.

```
virtual AcMapIE::ErrCode SetTargetCoordSys(  
    const ACHAR* pszCoordSysName  
) = 0;
```

Parameters	Description
pszCoordSysName	Input name of the coordinate system, as a Mentor abbreviation.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Coordinate System column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: SetUseUniqueKeyField Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets whether a unique key value is created for each incoming entity. This setting is ignored unless an existing or new object data table already has been set with SetDataMapping(). The key column has type integer. In the AutoCAD Map user interface, this setting corresponds to the Add Unique Key Field check box in the Attribute Data dialog box.

```
virtual AcMapIE::ErrCode SetUseUniqueKeyField(  
    const ACHAR* pszKeyFieldName  
) = 0;
```

Parameters	Description
pszKeyFieldName	Input name for key column. This name must follow the pre-Release 14 AutoCAD symbol-naming convention; that is, it can contain only alphanumeric, underscore, and hyphen characters. Multibyte characters are allowed.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadFieldName if the column name is invalid. Returns AcMapIE::ErrCode kErr\_Invalid if a column with the same name already exists as an input column. (This situation is not necessarily a problem, but it could render other import-related error reports and actions confusing.) Returns AcMapIE::ErrCode kErr\_ConflictWithTarget if the column name already exists as a mapped target column name. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer:: UseUniqueKeyField Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Determines whether a unique value is created for each incoming entity. This setting is ignored unless an existing or new object data table already has been set with `SetDataMapping()`. The key column has type integer. In the AutoCAD Map user interface, this setting corresponds to the Add Unique Key Field check box in the Attribute Data dialog box.

```
virtual bool UseUniqueKeyField(  
    const ACHAR*& pszKeyFieldName  
) const = 0;
```

Parameters	Description
pszKeyFieldName	Output name of the key column.

### Returns

Returns true if unique values are created; otherwise, returns false. If true, pszKeyFieldName column contains a non-empty string; if false, pszKeyFieldName contains an empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIEInputLayerIterator Class

[Classes](#)

An iterator over a collection of AcMapIEInputLayerinstances. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIEInputLayerIterator;
```

File

AcMapIEInputLayerIterator.h

☐ Methods



[~AcMapIEInputLayerIterator](#) Destroys an instance of this class.

[Done](#) Determines whether the iterator has reached the end of the collection.

[Find](#) Searches the collection for an element.

[Get](#) Retrieves the current element in the iteration.

[Rewind](#) Moves to the first element in the iteration.

[Step](#) Advances to the next element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)

[AcMapIEInputLayerIterator:: Find Method](#)

[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Searches the collection for an element.

```
virtual AcMapIEInputLayer* Find(  
    const ACHAR * pInputLayerName  
) = 0;
```

Parameters

Description

pInputLayerName

Input name of the input layer to find.

Returns

Returns the named AcMapIEInputLayer if found; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapIENameValueIterator Class

[Classes](#)

An iterator over a collection of name-value pairs. The case-insensitive name component of a name-value pair is a key value and must be unique within the collection. This class (which is similar to the STL map class) differs from the [AcMapIEExpressionTargetIterator](#) class in that this class has more lenient naming rules for the name component of a name-value pair. Do not explicitly delete any strings returned by any of the functions or their output parameters in this class (but delete all other types of returned objects).

```
class AcMapIENameValueIterator;
```

File

AcMapIENameValueIterator.h

☐ Methods



[~AcMapIENameValueIterator](#) Destroys an instance of this class.

[Add](#)

Inserts a new element into the collection after the current element and advances the iterator to the new element. After adding a new element successfully, call `Rewind()` to iterate over the entire collection.

[Clear](#)

Removes all elements from the collection.

[Done](#)

Determines whether the iterator has reached the end of the collection.

[Find](#)

Searches the collection for an element.

[Get](#)

Retrieves the current element in the iteration.

[Remove](#)

Removes an element from the collection.

[Rewind](#)

Moves to the first element in the iteration.

[Set](#)

Sets the value of a name-value pair in the collection.

[Step](#)

Advances to the next element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

[AcMapIENamespaceValueIterator::Add Method](#)

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Inserts a new element into the collection after the current element and advances the iterator to the new element. After adding a new element successfully, call `Rewind()` to iterate over the entire collection.

```
virtual AcMapIE::ErrCode Add(  
    const ACHAR* pszName,  
    const ACHAR* pszValue  
) = 0;
```

Parameters	Description
pszName	Input name of the name-value pair to add.
pszValue	Input value of the name-value pair to add.

Returns

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_BadParams` if either input string is NULL or empty. Returns `AcMapIE::ErrCode kErr_BadFieldName` if the input name is invalid. Returns `AcMapIE::ErrCode kErr_Fail` if the case-insensitive input name already exists in the collection.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

[AcMapIENamespaceValueIterator::Find Method](#)

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Searches the collection for an element.

```
virtual bool Find(  
    const ACHAR*& pszValue,  
    const ACHAR * pszName  
) = 0;
```

Parameters	Description
pszValue	Output value corresponding to the input name if the name-value pair was found. An empty string indicates that the element was not found.
pszName	Input case-insensitive name of the name-value pair to find.

## Returns

Returns true if the element was found; otherwise, returns false. False indicates an invalid output value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

[AcMapIENamespaceValueIterator::Get Method](#)

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Retrieves the current element in the iteration.

```
virtual bool Get(  
    const ACHAR*& pszName,  
    const ACHAR*& pszValue  
) = 0;
```

Parameters	Description
pszName	Output name of the current name-value pair.
pszValue	Output value of the current name-value pair.

Returns

Returns true if the current element is valid, or false if the iteration has no more elements to retrieve.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

[AcMapIENamespaceValueIterator::Remove Method](#)

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Removes an element from the collection.

```
virtual bool Remove(  
    const ACHAR* pszName  
) = 0;
```

Parameters

Description

pszName

Input case-insensitive name of the name-value pair to remove.

Returns

Returns true if the element was removed successfully; otherwise, returns false.

Remarks

If the iterator is pointing to the removed element, it advances to the next element if possible.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

AcMapIENamespaceValueIterator::Set Method

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Sets the value of a name-value pair in the collection.

```
virtual AcMapIE::ErrCode Set(  
    const ACHAR* pszName,  
    const ACHAR* pszValue  
) = 0;
```

Parameters	Description
pszName	Input case-insensitive name of the name-value pair to set.
pszValue	Input new value for the name-value pair to set.
Returns	

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_BadParams if either input string is NULL or empty. Returns AcMapIE::ErrCode kErr\_BadFieldName if the input name is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the case-insensitive input name does not already exist in the collection.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Import-Export](#)

Functions

[Import-Export](#)

☐ Functions

[AcMapExporter](#)

Retrieves the AutoCAD Map exporter object, as a singleton. Only a single instance of `AcMapIEExporter` exists (do not delete it). Always complete an active export before switching the current drawing. This function will succeed whenever the command-line command `MAPEXPORT` can succeed in the AutoCAD Map user interface. The return value `NULL` usually indicates an installation problem.

[AcMapExportFormatIterator](#)

Retrieves the AutoCAD Map export format iterator.

[AcMapImporter](#)

Retrieves the AutoCAD Map importer object, as a singleton. Only a single instance of `AcMapIEImporter` exists (do not delete it). Always complete an active import before switching the current drawing. This function will succeed whenever the command-line command `MAPIIMPORT` can succeed in the AutoCAD Map user interface. The return value `NULL` usually indicates an installation problem.

[AcMapImportFormatIterator](#)

Retrieves the AutoCAD Map import format iterator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Functions](#)

AcMapExporter Function

[Functions](#)

Retrieves the AutoCAD Map exporter object, as a singleton. Only a single instance of AcMapIEExporter exists (do not delete it). Always complete an active export before switching the current drawing. This function will succeed whenever the command-line command MAPEXPORT can succeed in the AutoCAD Map user interface. The return value NULL usually indicates an installation problem.

```
AcMapIEExporter* AcMapExporter();  
File
```

AcMapIEGlobal.h

Returns

Returns a pointer to the exporter, or NULL if the exporter could not be retrieved. Do not explicitly delete this returned value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Functions](#)

AcMapExportFormatIterator Function

[Functions](#)

Retrieves the AutoCAD Map export format iterator.

[AcMapIEFormatIterator](#)\* AcMapExportFormatIterator();

File

AcMapIEGlobal.h

Returns

Returns a pointer to the [AcMapIEFormatIterator](#) export format iterator object, or NULL if AcMapExporter() returned NULL. Do not explicitly delete this returned value.

Remarks

This iterator is the same class as the import format iterator but is a different instance that provides information about the collection of export formats only.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Functions](#)

AcMapImporter Function

[Functions](#)

Retrieves the AutoCAD Map importer object, as a singleton. Only a single instance of AcMapIEImporter exists (do not delete it). Always complete an active import before switching the current drawing. This function will succeed whenever the command-line command MAPIMPORT can succeed in the AutoCAD Map user interface. The return value NULL usually indicates an installation problem.

```
AcMapIEImporter* AcMapImporter();
```

File

AcMapIEGlobal.h

Returns

Returns a pointer to the importer, or NULL if the importer could not be retrieved. Do not explicitly delete this returned value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Functions](#)

AcMapImportFormatIterator Function

[Functions](#)

Retrieves the AutoCAD Map import format iterator.

[AcMapIEFormatIterator](#)\* AcMapImportFormatIterator();

File

AcMapIEGlobal.h

Returns

Returns a pointer to the [AcMapIEFormatIterator](#) import format iterator object, or NULL if AcMapImporter() returned NULL. Do not explicitly delete this returned value.

Remarks

This iterator is the same class as the export format iterator but is a different instance that provides information about the collection of import formats only.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Map Book: Class-Based API

### ▣ Classes

[AcMapMbError](#)

[AcMapMbMapBook](#)

[AcMapMbMapBookManager](#)

[AcMapMbMapBookReactor](#)

[AcMapMbMapSheetLayoutSettings](#)

[AcMapMbMapSheetTemplate](#)

[AcMapMbTile](#)

[AcMapMbTileGenerator](#)

[AcMapMbTileGeneratorSettings](#)

[AcMapMbTileGeneratorSettingsArea](#)

[AcMapMbTileGeneratorSettingsGrid](#)

[AcMapMbTileGeneratorSettingsManual](#)

This is class AcMapMbError.

Map-book-management Map B

This is a collection of static fun  
Manage map books.

Base class used to notify the ap  
Book events within an AutoCA

Class AcMapMbMapSheetLayo  
defines a set of the settings to f  
layout from a template drawing

Map-book-management Map S

This is class AcMapMbTile.

Class AcMapMbTileGenerator  
base class which defines the int  
Tile Generators.

This is a list of values we handl  
the base class It is mostly for th  
maintanence of the settings clas

source Map Book Scale Scale I  
Tiling scheme DisableEmptyTi

IntersectEntites InteresectEntity

OverlapPercent Class

AcMapMbTileGeneratorSetting  
settings class for tile generati

Class AcMapMbTileGenerator:  
contains the added properties n  
tile generation.

tile generation.

This is a list of values we handl  
Grid Tiler.

This is a list of values we handl  
Manual Tiler Selection set of pl

AcMapMbTileGeneratorSetting  
contains the added properties n  
tile generation.

tile generation.

Class AcMapMbTileNameGen

[AcMapMbTileNameGenerator](#)

virtual base class which defines the Name Generators.

[AcMapMbTileNameGeneratorData](#)

Class AcMapMbTileNameGenerator inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) the interface for the Name Generator.

[AcMapMbTileNameGeneratorGrid](#)

Class AcMapMbTileNameGenerator inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) the interface for the Name Generator.

[AcMapMbTileNameGeneratorIndexer](#)

This class encapsulates a naming indexer.

[AcMapMbTileNameGeneratorSequence](#)

Class AcMapMbTileNameGenerator inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) the interface for the Name Generator.

[AcMapMbTileNameGeneratorSettings](#)

Class AcMapMbTileNameGeneratorSettings is the base settings class for name generators. It contains the properties needed by all derived classes. The user will not instantiate this class as the derived classes will override the properties needed by actual name generators. The Name Generators and the Name Generator Settings classes have a direct relationship. For any future expansion, users will instantiate both a new Generator and Settings class. The appropriate Generator will be overridden by the Settings class. The DECLARE\_OBJECT macro is used in the declaration of these classes to be a part of the ObjectAF tree.

[AcMapMbTileNameGeneratorSettingsData](#)

Class AcMapMbTileNameGeneratorSettingsData contains the added property needed for name generation.

[AcMapMbTileNameGeneratorSettingsGrid](#)

Class AcMapMbTileNameGeneratorSettingsGrid contains the added properties needed for name generation.

name generation.

Class

[AcMapMbTileNameGeneratorSettingsSequence](#)

AcMapMbTileNameGeneratorSettingsSequence contains the added property needed for sequential name generation.

[AcMapMbTileSet](#)

Provides functions to create, name, and manage the tiles used by a MapBook.

Links

[Map Book: Class-Based API](#)

Classes

[Map Book: Class-Based API](#)

▣ Classes

[AcMapMbError](#)

[AcMapMbMapBook](#)

[AcMapMbMapBookManager](#)

[AcMapMbMapBookReactor](#)

[AcMapMbMapSheetLayoutSettings](#)

[AcMapMbMapSheetTemplate](#)

[AcMapMbTile](#)

[AcMapMbTileGenerator](#)

[AcMapMbTileGeneratorSettings](#)

[AcMapMbTileGeneratorSettingsArea](#)

This is class AcMapMbError.

Map-book-management Map B

This is a collection of static fun  
Manage map books.

Base class used to notify the ap  
Book events within an AutoCA  
Derive custom reactors from th  
AcMapMBRecator class.

To add a reactor: Derive your c  
AcMapMBMyReactor from  
AcMapMbMapBookRecator.O  
virtual functions of this reactor

Class AcMapMbMapSheetLay  
defines a set of the settings to f  
layout from a template drawing

Map-book-management Map S

This is class AcMapMbTile.

Class AcMapMbTileGenerator  
base class which defines the int  
Tile Generators.

This is a list of values we handl  
the base class It is mostly for th  
maintenance of the settings clas  
source Map Book Scale Scale I  
Tiling scheme DisableEmptyTi  
IntersectEntites InteresectEntity  
OverlapPercent Class

AcMapMbTileGeneratorSetting  
settings class for tile generati

Class AcMapMbTileGenerator!  
contains the added properties n  
tile generation.

[AcMapMbTileGeneratorSettingsGrid](#)

This is a list of values we handle  
Grid Tiler.

[AcMapMbTileGeneratorSettingsManual](#)

This is a list of values we handle  
Manual Tiler Selection set of pl  
AcMapMbTileGeneratorSetting  
contains the added properties n  
tile generation.

[AcMapMbTileNameGenerator](#)

Class AcMapMbTileNameGen  
virtual base class which defines  
the Name Generators.

[AcMapMbTileNameGeneratorData](#)

Class AcMapMbTileNameGen  
inherits from the pure virtual ba  
[AcMapMbTileNameGenerator](#)  
the interface for the Name Gen

[AcMapMbTileNameGeneratorGrid](#)

Class AcMapMbTileNameGen  
inherits from the pure virtual ba  
[AcMapMbTileNameGenerator](#)  
the interface for the Name Gen

[AcMapMbTileNameGeneratorIndexer](#)

This class encapsulates a namir

[AcMapMbTileNameGeneratorSequence](#)

Class AcMapMbTileNameGen  
inherits from the pure virtual ba  
[AcMapMbTileNameGenerator](#)  
the interface for the Name Gen

[AcMapMbTileNameGeneratorSettings](#)

Class AcMapMbTileNameGen  
the base settings class for name  
Class

[AcMapMbTileNameGeneratorSettingsData](#)

AcMapMbTileNameGenerator!  
contains the added property nee  
name generation.

[AcMapMbTileNameGeneratorSettingsGrid](#)

Class  
AcMapMbTileNameGenerator!  
contains the added properties n  
name generation.

[AcMapMbTileNameGeneratorSettingsSequence](#)

Class  
AcMapMbTileNameGenerator!  
contains the added property nee  
sequential name generation.

## [AcMapMbTileSet](#)

Provides functions to create, na  
the tiles used by a MapBook.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbError Class

[Classes](#)

This is class AcMapMbError.

```
class AcMapMbError;
```

File

AcMapMbError.h

☐ Enumerations

📄 [EMbStatus](#)

This enum represents the list of defined Map Book errors.

📄 [ErrorClass](#)

KMapMbClassErrors represents a type of errors associated with Map Book application.

☐ Methods

🔷 [~AcMapMbError](#)

Destroys an instance of this class.

🔷 [AcMapMbError](#)

Constructs an instance of this class.

[IsError](#)

Checks if the status code indicates an error.

[IsSuccess](#)

Checks if the status code indicates success.

[IsWarning](#)

Checks if the status code indicates a warning.

[Message](#)

Loads an error message associated with the error code from resource file.

[PushErrorMessage](#)

Pushes an error entry to the error stack.

[PushErrorMessage](#)

Pushes an error entry to the error stack.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbError Class](#), [AcMapMbError Class](#)  
AcMapMbError:: PushErrorMessage Method  
[AcMapMbError Class](#) | [AcMapMbError Class](#)

Pushes an error entry to the error stack.

```
static void PushErrorMessage(  
    EMbStatus es,  
    const ACHAR* pszParam  
);
```

Parameters	Description
es	Input error status
pszParam	Input string error parameter

Returns

Nothing.

Remarks

The entry includes kMapMbClassErrors as error type, es as an error code and the error message made from pszParam and format string associated with es if any.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbMapBook Class

[Classes](#)

Map-book-management Map Book class.

```
class AcMapMbMapBook : public AcDbObject;  
File
```

AcMapMbMapBook.h

Remarks

These objects are stored in the drawing in an AcDbDictionary and managed by the MapBookManager.

☐ Enumerations

 [EModificationType](#)

Bit flags indicating to a reactor what kind of modification is taking place.

☐ Methods



[~AcMapMbMapBook](#)

Destroys an instance of this class.



[AcMapMbMapBook](#)

Constructs an instance of this class.

[dwgInFields](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also dxfOutFields()

<a href="#"><u>dxfoutFields</u></a>	in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>GetCoordSysName</u></a>	Retrieves the coordinate system name string.
<a href="#"><u>GetNorthDirection</u></a>	Retrieves the north direction.
<a href="#"><u>GetScaleString</u></a>	Retrieves the scale string.
<a href="#"><u>GetSettings</u></a>	Gets the Map Sheet Layout Settings used to generate the Map Book. .
<a href="#"><u>GetSettings</u></a>	Gets the Tile Generator settings used to generate the Map Book.
<a href="#"><u>GetSettings</u></a>	Gets the Tile Name Generator settings used to generate the Map Book.
<a href="#"><u>GetTileCount</u></a>	Retrieves the number of tiles.
<a href="#"><u>Implementation</u></a>	Returns the implementation object.
<a href="#"><u>IsSaveRequired</u></a>	Checks if the Map Book in the current configuration has been saved so that it can be published.
<a href="#"><u>IsSheetSetValid</u></a>	Checks if a referred Sheet Set file exists and has been created not later than the drawing file.
<a href="#"><u>Name</u></a>	Retrieves the name of this Map Book.
<a href="#"><u>PublishToDwf</u></a>	Publishes the specified tiles to the DWF file, if there are no tiles specified, publishes the entire Map Book.
<a href="#"><u>PublishToPlotter</u></a>	Publishes the specified tiles to the plotter specified by the configuration file.
<a href="#"><u>RebuildSheetSet</u></a>	Recreates a Sheet Set.
<a href="#"><u>ResolveSheetSet</u></a>	Resets a reference to a Sheet Set and refreshes sheets with the proper drawing location.
<a href="#"><u>SetCoordSysName</u></a>	Sets the coordinate system name.
<a href="#"><u>SetName</u></a>	Sets the name of the map Book.
<a href="#"><u>SetNorthDirection</u></a>	Sets the north direction.
<a href="#"><u>SetScaleString</u></a>	Sets the scale string.
<a href="#"><u>SheetSetFileName</u></a>	Retrieves the file name of the Map Sheet Set associated with this MapBook.
<a href="#"><u>SheetSetSubsetName</u></a>	Retrieves the Subset name within the Map Sheet Set if any.
	Shows/hides grid objects (polylines) associated with

[ShowGridObjects](#)

the Tile Set.

[subClose](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[subErase](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[TileSet](#)

Gets the Tile Set of the Map Book.

[wblockClone](#)

Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object pOwnerObject. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: GetCoordSysName Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrieves the coordinate system name string.

```
Acad::ErrorStatus GetCoordSysName(  
    ACHAR*& pszCoordSysName  
) const;
```

Parameters	Description
dCoordSysName	Output the string representing coordinate system description string.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as Coordinate System custom property for the purpose of displaying the description as a field on the layouts. This value can be set by the user thru the sheet set manager UI.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: GetScaleString Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrieves the scale string.

```
Acad::ErrorStatus GetScaleString(  
    ACHAR*& pszScaleStr  
) const;
```

Parameters	Description
pszScaleStr	Output a scale string.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as Scale custom property in "1:XXX" format; the initial value is generated from the actual layout format. Changing this property does not alter the Map Book, it just changes a Sheet Set property.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: IsSaveRequired Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Checks if the Map Book in the current configuration has been saved so that it can be published.

```
bool IsSaveRequired() const;
```

Returns

Returns true if the drawing has not been saved since the Map Book has been created or rebuilt.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: IsSheetSetValid Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Checks if a referred Sheet Set file exists and has been created not later than the drawing file.

```
bool IsSheetSetValid() const;
```

Returns

Returns true if the .DST file is valid, false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: PublishToDwf Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Publishes the specified tiles to the DWF file, if there are no tiles specified, publishes the entire Map Book.

```
Acad::ErrorStatus PublishToDwf(  
    const AcArray<AcMapMbMapBookManager::TILEID>& tileIds,  
    const ACHAR* pszFileName  
);
```

Parameters	Description
tileIds	Input Ids of the tiles to be published; if the array is empty, the entire Map Book will be published.
pszFileName	Input the name of the DWF file to be created; if the file with this name already exists it will be overwritten.

## Returns

Returns Acad::eOk if successful; returns Acad::eDwgNeedsAFullSave if drawing must be saved; Acad::eNotCurrentDatabase in the case the Sheet Set is out of synch with the Map Book otherwise, returns a different error code.

## Remarks

If the database has changed, the drawing must be saved before publishing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: PublishToPlotter Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Publishes the specified tiles to the plotter specified by the configuration file.

```
Acad::ErrorStatus PublishToPlotter(  
    const AcArray<AcMapMbMapBookManager::TILEID>& tileIds,  
    const ACHAR* pszConfigFileName = NULL  
);
```

Parameters	Description
tileIds	Input Ids of the tiles to be published; if the array is empty, the entire Map Book will be published.
pszConfigFileName	Input the name of the plot configuration file; if not specified, the current plotter settings associated with the Sheet will be used.

## Returns

Returns Acad::eOk if successful; returns Acad::eDwgNeedsAFullSave if drawing must be saved; Acad::eNotCurrentDatabase in the case the Sheet Set is out of synch with the Map Book otherwise, returns a different error code.

## Remarks

If there are no tiles specified, publishes the entire Map Book. If the configuration file is not specified, the current plotter settings associated with the Sheet will be used. If the database has changed, the drawing must be saved before publishing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: RebuildSheetSet Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Recreates a Sheet Set.

```
Acad::ErrorStatus RebuildSheetSet(  
    const TCHAR* pPathName = NULL  
);
```

Parameters	Description
pPathName	Passed-in a .DST path name. If NULL use the existing path.

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook::ResolveSheetSet Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Resets a reference to a Sheet Set and refreshes sheets with the proper drawing location.

```
Acad::ErrorStatus ResolveSheetSet(  
    const ACHAR* pszFileName  
);
```

Parameters	Description
pszFileName	Input Sheet Set file name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The specified file should be the sheet set generated with the Map Book. The sheet set manager has the ability to find a given sheet set when that sheet set has been moved to a different location in the file system. See sheet set manager documentation on this feature.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: SetCoordSysName Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Sets the coordinate system name.

```
Acad::ErrorStatus SetCoordSysName(  
    const ACHAR* pszCoordSysName  
);
```

Parameters	Description
dCoordSysName	Input the string representing coordinate system description string.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as CoordinateSystem custom property. Changing this Map Book property does not change the actual coordinate system associated with the drawing file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook::SetName Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Sets the name of the map Book.

```
Acad::ErrorStatus SetName(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input Map Book name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: SetScaleString Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Sets the scale string.

```
Acad::ErrorStatus SetScaleString(  
    const ACHAR* pszScaleStr  
);
```

Parameters

Description

pszScaleStr

Input a scale string; it is up to an application what string to specify, but in general it should be a string in "1:XXX" format.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as Scale custom property.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbMapBookManager Class

[Classes](#)

This is a collection of static functions which Manage map books.

```
class AcMapMbMapBookManager ;  
File
```

AcMapMbMapBookManager.h

Remarks

There may be several Map Books in a drawing. They are resident in the drawing.

☐ Enumerations

 [EAdjacentTile](#) Defines a direction for selecting adjacent tiles.

☐ Methods

<a href="#">AddMapBookCreationReactor</a>	Adds a map book creation reactor to monitor map book creation process such as end of tiling and sheet creation.
<a href="#">AddMapBookReactor</a>	Adds a map-book-management reactor to monitor map book management related activities.
<a href="#">GenerateMapBook</a>	Generates new Map Book based on the settings supplied.
<a href="#">GetAt</a>	Retrieves the ID of the named map book.
<a href="#">GetAt</a>	Retrieves the named map book.
<a href="#">GetCurrent</a>	Retrieves the current Map Book.
<a href="#">GetMapBookForLayout</a>	Retrieves an Id of a Map Book associated with the given layout if any.
<a href="#">Has</a>	Checks if a Map Book with the specified name has already been defined within a drawing database.
	Determines whether a project has map book

<a href="#">HasMapBookData</a>	data.
<a href="#">LoadSettings</a>	Fills appropriate settings objects and populates them from the specified XML file. Returns Acad::eOk if successful; otherwise, returns a different error code.
<a href="#">mbMapBookManagerImp</a>	Returns the implementation object.
<a href="#">NewIterator</a>	Returns a new map book dictionary iterator.
<a href="#">Remove</a>	Removes the map book with the specified ID.
<a href="#">RemoveMapBookCreationReactor</a>	Removes a map book creation reactor.
<a href="#">RemoveMapBookReactor</a>	Removes a map book reactor from a database.
<a href="#">ResetName</a>	Renames the map book with the specified ID.
<a href="#">ResetTileView</a>	Resets the tile view for the specified tile.
<a href="#">SaveSettings</a>	Save the specified map book settings in the specified XML file.
<a href="#">SetCurrent</a>	Sets the current Map Book.
<a href="#">SetMapBookDMEnvironment</a>	Sets the Display Management environment such as DM Map and Scale associated with the Map Book.
<a href="#">SetMapBookEnvironment</a>	Sets the environment associated with the Map Book.
<a href="#">UpdateMapBook</a>	Modifies existing Map Book based on the settings supplied.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager::GenerateMapBook Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Generates new Map Book based on the settings supplied.

```
static Acad::ErrorStatus GenerateMapBook(  
    AcDbObjectId& Id,  
    AcDbDatabase* pDatabase,  
    const ACHAR* pszName,  
    const AcMapMbTileGeneratorSettings* pTileGeneratorSettings,  
    const AcMapMbTileNameGeneratorSettings* pTileNameGeneratorSettings,  
    const AcMapMbMapSheetLayoutSettings* pLayoutTemplateSettings,  
    const ACHAR* pszSheetSetFileName,  
    const ACHAR* pszSubsetName = NULL  
);
```

Parameters	Description
Id	Output ID of the created Map Book object.
pDatabase	Input drawing database.
pszName	Input name of the map.
pTileGeneratorSettings	Input tile generator settings object. The user will create the correct settings object and set the appropriate values. It is never correct to use a base class settings object.
pTileNameGeneratorSettings	Input tile name generator settings object. The user will create the correct settings object and set the appropriate values. It is never correct to use a base class settings object.
pLayoutTemplateSettings	Input layout template settings object. The user will create the correct settings object and set the appropriate values.
pszSheetSetFileName	Input Sheet Set file name, if the file with this name does not exist, it will be created. If the file already exists it will be overwritten.
pszSubsetName	Input category name within the Map Sheet Set. The category name includes the full path where a slash "/" is used as a delimiter, for example: "Root/Nested_Subset/Map_Book_category". If

this Sheet Set category already exists, it will be overwritten.

## Returns

Returns Acad::Ok if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager:: GetAt Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Retrieves the ID of the named map book.

```
static Acad::ErrorStatus GetAt(  
    AcDbObjectId& Id,  
    AcDbDatabase* pDatabase,  
    const ACHAR* pszName  
);
```

Parameters	Description
Id	Output ID of the map book.
pDatabase	Input pointer to a drawing database.
pszName	Input name of the map book.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: GetAt Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Retrieves the named map book.

```
static Acad::ErrorStatus GetAt(  
    AcMapMbMapBook*& pMapBook,  
    AcDbDatabase* pDatabase,  
    const ACHAR* pszName,  
    AcDb::OpenMode mode  
);
```

Parameters	Description
pMapBook	Output AcMapMbMapBook.
pDatabase	Input pointer to a drawing database.
pszName	Input name of the map book.
mode	Input mode for opening the map book.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

AcMapMbMapBookManager:: Has Method

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Checks if a Map Book with the specified name has already been defined within a drawing database.

```
static bool Has(  
    const ACHAR* pszName,  
    AcDbDatabase* pDatabase  
);
```

Parameters	Description
pszName	Input name of the map book.
pDatabase	Input pointer to a drawing database.

Returns

Returns true if a Map Book with the specified name is already defined; false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager::LoadSettings Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Fills appropriate settings objects and populates them from the specified XML file.

Returns Acad::eOk if successful; otherwise, returns a different error code.

```
static Acad::ErrorStatus LoadSettings(  
    AcArray<AcDbObject*>& SettingsArray,  
    const ACHAR* pFileName = NULL  
);
```

Parameters	Description
pszFileName	Input name of the XML file. This should be a fully qualified file name.
pSettingsArray	Output settings array.

## Remarks

This function may push errors onto the AutoCAD Map Error Stack. Typically the user will add no more than one of each derived type of settings object to the array. Valid objects can be expected only if an object of that type was saved previously.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager:: ResetName Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Renames the map book with the specified ID.

```
static Acad::ErrorStatus ResetName(  
    const AcDbObjectId& mapId,  
    const ACHAR* pszNewName  
);
```

Parameters	Description
mapId	Input ID of the map book to rename.
pszNewName	Input new map book name.

## Returns

Returns Acad::eOk if successful; returns Acad::eDuplicateKey if there is another map book with this name returns Acad::eInvalidInput if the name is not a valid Extended Symbol name or if the name is the same; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager::ResetTileView Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Resets the tile view for the specified tile.

```
static bool ResetTileView(  
    AcMapMbMapBookManager::TILEID tileId,  
    const AcDbObjectId& MapBookId = NULL  
);
```

Parameters	Description
tileId	Input tile Id.
MapBookId	Input Map Book Id, curren Map Book by default.

Returns

Returns true if the project has map book data.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: SaveSettings Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Save the specified map book settings in the specified XML file.

```
static Acad::ErrorStatus SaveSettings(  
    AcArray<AcDbObject*>& SettingsArray,  
    const ACHAR* pFileName  
);
```

Parameters	Description
pszFileName	Input name of the XML file. This should be a fully qualified file name.
pSettingsArray	Input array of valid settings objects.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This function may push errors onto the AutoCAD Map Error Stack. Typically the user will save settings objects of at most 1 of each type of derived settings objects.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

AcMapMbMapBookManager::UpdateMapBook Method

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Modifies existing Map Book based on the settings supplied.

```
static Acad::ErrorStatus UpdateMapBook(  
    const AcDbObjectId& Id,  
    const AcMapMbTileGeneratorSettings* pTileGeneratorSettings,  
    const AcMapMbTileNameGeneratorSettings* pTileNameGeneratorSettings,  
    const AcMapMbMapSheetLayoutSettings* pLayoutTemplateSettings,  
    const ACHAR* pszSheetSetFileName = NULL,  
    const ACHAR* pszSubsetName = NULL  
);
```

Parameters	Description
Id	Input ID of the Map Book object to modify.
pTileGeneratorSettings	Input tile generator settings object. The user will create the correct settings object and set the appropriate values. It is never correct to use a base class settings object. It is not necessary to use the same type settings object as when the MapBook was created.
pTileNameGeneratorSettings	Input tile generator settings object. The user will create the correct settings object and set the appropriate values. It is never correct to use a base class settings object. It is not necessary to use the same type settings object as when the MapBook was created.
pLayoutTemplateSettings	Input layout template settings object. The user may edit values from previous settings.
pszSheetSetFileName	Input Sheet Set file name, if the file with this name does not exist, it will be created. NULL value indicates that previous sheet set data will be overwritten.
pszSubsetName	Input category name within the Map Sheet Set. The category name includes the full path where a slash "/" is used as a delimiter, for example: "Root/Nested_Subset/Map_Book_category". If

this Sheet Set already exists, it will be overwritten

pszName

Input name of the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbMapBookReactor Class

[Classes](#)

Base class used to notify the application of Map Book events within an AutoCAD Map project. Derive custom reactors from the AcMapMBReactor class.

To add a reactor: Derive your class AcMapMBMyReactor from AcMapMbMapBookReactor. Override the virtual functions of this reactor base class..

```
class AcMapMbMapBookReactor;
```

File

AcMapMbMapBookReactor.h

☐ Methods

[MapBookAppended](#)

Invoked when a new map book is appended or after a reactor is attached to a project that has map book(s) defined.

[MapBookErased](#)

Invoked when a map book is detached or erase operation is undone.

[MapBookModified](#)

Invoked when a map book is detached or erase operation is undone.

[MapBookSetCurrent](#)

Invoked when a current Map Book gets set or at the very beginning of the Map session.

[MapBookTileModified](#)

Invoked when a map book tile is modified.

[MapBookTileWillBeErased](#)

Invoked when a map book tile is about to be erased.

[MapBookTreeNodeModified](#)

Invoked when a map book tree node is modified.

[MapBookWillBeErased](#)

Invoked when a map book is about to be erased or erase operation is about to be undone.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbMapSheetLayoutSettings Class

[Classes](#)

Class AcMapMbMapSheetLayoutSettings defines a set of the settings to form Map Sheet layout from a template drawing.

```
class AcMapMbMapSheetLayoutSettings : public AcDbObject;
```

File

AcMapMbMapSheetLayoutSettings.h

☐ Enumerations

 [EMapKeyType](#) Enumerates supported types of a Map Key.

 [EMapLegendType](#) Enumerates supported types of legend

☐ Methods



[~AcMapMbMapSheetLayoutSettings](#) Destroys an instance of this class.



[AcMapMbMapSheetLayoutSettings](#) Constructs an instance of this class.

[AdjacentArrowBlockName](#) Retrieves an adjacent arrow block name.

[AllocateMapSheetTemplate](#) Allocates an instance of Map Sheet Template class.

[clone](#) Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object read its data. See also dwgInFields() in the AutoCAD

[dwgInFields](#)

ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[GetLegendBox](#)

Retrieves a Legend box.

[GetMapKeyLinkDrawing](#)

Retrieves a Display Manager Map and threshold scale value to associate a link drawing with the Linked Drawing Map Key.

[Implementation](#)

Returns the implementation object.

[IncludeAdjacentArrows](#)

Retrieves the flag to include a adjacent arrows to a Map Sheet layout.

[IncludeTitleBlock](#)

Retrieves the flag to include a title block to a Map Sheet layout.

[LegendType](#)

Retrieves a Legend type.

[MapKeyExternalFileName](#)

Retrieves a drawing file name for the External Drawing Map Key.

[MapKeyLayerPattern](#)

Retrieves a pattern to select layers visible within a current drawing Map Key.

<a href="#"><u>MapKeyType</u></a>	Retrieves a Map Key type.
<a href="#"><u>SetAdjacentArrowBlockName</u></a>	Sets an adjacent arrow block name.
<a href="#"><u>SetIncludeAdjacentArrows</u></a>	Sets the flag to include adjacent arrows to a Map Sheet layout.
<a href="#"><u>SetIncludeTitleBlock</u></a>	Sets the flag to include a title block to a Map Sheet layout.
<a href="#"><u>SetLegendBox</u></a>	Sets a Legend box.
<a href="#"><u>SetLegendType</u></a>	Sets a Legend type.
<a href="#"><u>SetMapKeyExternalFileName</u></a>	Sets a drawing file name for the External Drawing Map Key.
<a href="#"><u>SetMapKeyLayerPattern</u></a>	Sets a pattern to select layers visible within a current drawing Map Key.
<a href="#"><u>SetMapKeyLinkDrawing</u></a>	Sets a Display Manager Map and threshold scale value to associate a link drawing with the Linked Drawing Map Key.
<a href="#"><u>SetMapKeyType</u></a>	Sets a Map Key type.
<a href="#"><u>SetTemplateFileName</u></a>	Sets a template file name.
<a href="#"><u>SetTemplateLayoutName</u></a>	Sets a template layout name.
<a href="#"><u>SetTitleBlockName</u></a>	Sets a title block name.
<a href="#"><u>TemplateFileName</u></a>	Retrieves a template file name.
<a href="#"><u>TemplateLayoutName</u></a>	Retrieves a template layout name.
<a href="#"><u>TitleBlockName</u></a>	Retrieves a title block name.
<a href="#"><u>xmlInFields</u></a>	Lets this object read its data. See also <code>dwgInFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>xmlOutFields</u></a>	Lets this object write its data. See also <code>dwgOutFields()</code> in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: GetMapKeyLinkDrawing Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a Display Manager Map and threshold scale value to associate a link drawing with the Linked Drawing Map Key.

```
Acad::ErrorStatus GetMapKeyLinkDrawing(  
    const ACHAR*& pszMapName,  
    double& dThresholdScale  
) const;
```

Parameters	Description
pszMapName	Output Display Manager Map name.
dThresholdScale	Output Display Manager Map Threshold Scale value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetAdjacentArrowBlockName Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets an adjacent arrow block name.

```
Acad::ErrorStatus SetAdjacentArrowBlockName(  
    const ACHAR* pszBlockName  
);
```

Parameters	Description
pszBlockName	Input adjacent arrow block name. This block has to be defined within a template file specified via SetTemplateName()

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

[AcMapMbMapSheetLayoutSettings:: SetMapKeyExternalFileName Method](#)  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a drawing file name for the External Drawing Map Key.

```
Acad::ErrorStatus SetMapKeyExternalFileName(  
    const ACHAR* pszFileName  
);
```

Parameters	Description
pszFileName	Input file name. The name should specify a complete path name with the .DWG extension. The function will validate to insure the file exists.

## Returns

Returns Acad::eOk if successful; Acad::eFilerError if the file cannot be found; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetMapKeyLayerPattern Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a pattern to select layers visible within a current drawing Map Key.

```
Acad::ErrorStatus SetMapKeyLayerPattern(  
    const ACHAR* pszLayerPattern  
);
```

Parameters	Description
pszLayerPattern	Input a pattern to select layers visible within a current drawing Map Key. The pattern relies on acdbSNValid() to select layers.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The patterns will be comma delimited, and may contain the regular expressions defined for acutWcMatch.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetMapKeyLinkDrawing Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a Display Manager Map and threshold scale value to associate a link drawing with the Linked Drawing Map Key.

```
Acad::ErrorStatus SetMapKeyLinkDrawing(  
    const ACHAR* pszMapName,  
    double dThresholdScale  
);
```

Parameters	Description
pszMapName	Input Display Manager Map name. The Display Manager Map with this name should have been defined in the current drawing.
dThresholdScale	Input Display Manager Map Threshold Scale value. The specified Display Manager map should have this Threshold Scale value defined.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetTemplateFileName Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a template file name.

```
Acad::ErrorStatus SetTemplateFileName(  
    const ACHAR* pszFileName  
);
```

Parameters	Description
pszFileName	Input file name. The name should specify a complete path name with the .DWT or .DWG file extension. This method validates that a file with the specified name exists.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

A layout from this template will be used as a prototype for the Map Sheet layout.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetTemplateLayoutName Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a template layout name.

```
Acad::ErrorStatus SetTemplateLayoutName(  
    const ACHAR* pszLayoutName  
);
```

Parameters	Description
pszLayoutName	Input template layout name. A template with this name should exist within a template file.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings::SetTitleBlockName Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a title block name.

```
Acad::ErrorStatus SetTitleBlockName(  
    const ACHAR* pszBlockName  
);
```

Parameters	Description
pszBlockName	Input title block name. This block has to be defined within a template file specified via SetTemplateFileName(). Setting the title block name switches it on, i.e. IncludeTitleBlock() will return true.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: xmlInFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: xmlOutFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbMapSheetTemplate Class

[Classes](#)

Map-book-management Map Sheet Template.

```
class AcMapMbMapSheetTemplate;  
File
```

AcMapMbMapSheetTemplate.h

Remarks

This class provides a wrapper for the Template drawing file and provides helper functions as needed to access this file.

☐ Enumerations



[ETemplateElementType](#) Defines template elements.

☐ Methods



[~AcMapMbMapSheetTemplate](#) Destroys an instance of this class.



[AcMapMbMapSheetTemplate](#) Constructs an instance of this class.

[AdjacentEnumToString](#) Gets the key string for the specified direction.

[AdjacentStringToEnum](#) Gets the adjacent direction for the specified key string value.

[CloneLayout](#) Clones the layout specified by the settings object from the template drawing to the target drawing.

[Close](#) Closes the template drawing file.

[DwgDatabase](#) Retrieves the drawing database object associated with the Map Sheet template.

[GetAdjacentArrow](#) Gets the adjacent arrow Id for the specified direction from the layout specified by the settings object.

Gets the adjacent arrow Ids for layout specified

[GetAdjacentArrows](#)

by the settings object.

[GetArrowDirection](#)

Gets adjacency direction from the adjacent arrow placeholder.

[GetBlockNames](#)

Retrieves the list of block names defined within the specified Map Sheet template drawing.

[GetElement](#)

Retrieves the Id of the specified template element.

Returns the object id of the object in the specified layout that is marked as an element of the specified type.

[GetElement](#)

Returns Acad::eOk if successful; Returns Acad::eAmbiguousOutput if there is more than one element of the specified type. Returns Acad::eKeyNotFound if there are no elements of the specified type; otherwise, returns a different error code.

Returns an array of object ids of the objects in the specified layout that are marked as an element of the specified type.

[GetElement](#)

Returns Acad::eOk if successful; Returns Acad::eKeyNotFound if there are no entities of the specified type; otherwise, returns a different error code.

[GetLayoutNames](#)

Retrieves the list of layout names defined within the specified Map Sheet template drawing.

[GetTileSize](#)

Gets height and width of tile from the Map viewport.

[Implementation](#)

Returns the implementation object.

[IsElement](#)

Checks if the object specified by the Id is an element of the specified type.

[LayoutId](#)

Retrieves current template layout Id.

[MarkElement](#)

Marks the selected template object with XData so that it is considered as a template element of the specified type.

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Open

Opens the template drawing file for reading. Remove marker XData from the selected template object so that it is considered as a template element any more.

### UnMarkElement

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Validate

Validates if the specified layout can be used as a Map Sheet layout.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate::AdjacentEnumToString Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets the key string for the specified direction.

```
static Acad::ErrorStatus AdjacentEnumToString(  
    const ACHAR*& pszSting,  
    AcMapMbMapBookManager::EAdjacentTile adjDirection  
);
```

Parameters	Description
pszSting	Output key string
adjDirection	Input adjacent direction

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate::AdjacentStringToEnum Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets the adjacent direction for the specified key string value.

```
static Acad::ErrorStatus AdjacentStringToEnum(  
    AcMapMbMapBookManager::EAdjacentTile & adjDirection,  
    const ACHAR* pszSting  
);
```

Parameters	Description
adjDirection	Output adjacent direction
pszSting	Input key string

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: CloneLayout Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Clones the layout specified by the settings object from the template drawing to the target drawing.

```
virtual Acad::ErrorStatus CloneLayout(  
    AcDbDatabase* pTargetDb,  
    const ACHAR* pszDestinationName,  
    AcDbIdMapping& idMapping  
) const;
```

Parameters	Description
pTargetDb	Input target drawing database.
pszDestinationName	Input name to be assigned to the cloned layout in the destination database.
idMapping	Input after cloning contains mapping between template database and destination database Ids.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The specified name gets assigned to the layout in the destination drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
[AcMapMbMapSheetTemplate:: GetBlockNames Method](#)  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Retrieves the list of block names defined within the specified Map Sheet template drawing.

```
static Acad::ErrorStatus GetBlockNames(  
    AcMapStringArray& aBlockNames,  
    const ACHAR* pszFileName  
);
```

Parameters	Description
aBlockNames	Output array of block names defined within a drawing file. Anonymous blocks are filtered out from this list.
pszFileName	Input file name. The name should specify a complete path name with the .DWT or .DWG extension. This method validates if a file with the specified name exists.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
[AcMapMbMapSheetTemplate:: GetLayoutNames Method](#)  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Retrieves the list of layout names defined within the specified Map Sheet template drawing.

```
static Acad::ErrorStatus GetLayoutNames(  
    AcMapStringArray& aLayoutNames,  
    const ACHAR* pszFileName  
);
```

Parameters	Description
aLayoutNames	Output array of layout names defined within a drawing file.
pszFileName	Input file name. The name should specify a complete path name with the .DWT or .DWG extension. This method validates if a file with the specified name exists.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: Validate Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Validates if the specified layout can be used as a Map Sheet layout.

```
static bool Validate(  
    const ACHAR * pszFileName,  
    const ACHAR* pszLayoutName  
);
```

Parameters	Description
pszFileName	Input file name. The name should specify a complete path name with the .DWT or .DWG extension. This method validates if a file with the specified name exists and that the specified layout exists within that file.
pszLayoutName	Input layout name

## Returns

Returns true if this is a valid template layout; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTile Class

[Classes](#)

This is class AcMapMbTile.

```
class AcMapMbTile : public AcRxObject;  
File
```

AcMapMbTile.h

☐ Methods

<a href="#">~AcMapMbTile</a>	Destroys an instance of this class.
<a href="#">AcMapMbTile</a>	Constructs an instance of this class.
<a href="#">AcMapMbTile</a>	Constructs an instance of this class.
<a href="#">AcMapMbTile</a>	Copies an instance of this class.
<a href="#">AddHyperlink</a>	Assigns a hyperlink to this tile to the specified adjacent arrow object.
<a href="#">Center</a>	Returns the center of a tile.
<a href="#">GetAdjacentTile</a>	Gets the adjacent tile of the specified direction for the tile.
<a href="#">GetAdjacentTile</a>	Gets the adjacent tile of the specified direction for the tile.
<a href="#">GetLayoutIdAt</a>	Returns the layout object id at the index.
<a href="#">GetTileSize</a>	Returns the size of the tile.
<a href="#">GetViewPortSize</a>	Gets the size of the tile as viewed.
<a href="#">Implementation</a>	Returns the implementation object.
<a href="#">IsEnabled</a>	Gets whether the tile is enabled.
<a href="#">Name</a>	Returns the name of the tile.
<a href="#">OnTileErased</a>	This is a notification that every tile except the erased one receives.
<a href="#">RefObject</a>	Returns the reference object associated with the tile.
<a href="#">SetAdjacentTile</a>	Sets the adjacent tile of the specified direction for the tile.

<a href="#"><u>SetName</u></a>	Sets a tile name.
<a href="#"><u>SheetHandle</u></a>	Returns the unique handle of the Sheet in the Sheet Set.
<a href="#"><u>StreamIn</u></a>	Reads data from a dwg file.
<a href="#"><u>StreamIn</u></a>	Reads data from a dxf file.
<a href="#"><u>StreamOut</u></a>	Loads data into a dwg file.
<a href="#"><u>StreamOut</u></a>	Loads data into a dxf file.
<a href="#"><u>TileId</u></a>	Returns the unique tile Id for this tile.
<a href="#"><u>TileSet</u></a>	Returns the pointer to the Tile Set it belongs to.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

AcMapMbTile::SetName Method

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Sets a tile name.

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR* pszName  
);
```

Parameters

Description

pszName

Input tile name. The name should specify a valid tile name. Tile names follow Extended Symbol Name rules. Many AutoCAD symbols such as Layer Names follow these rules.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Returns Acad::eNullPtr if pszName is NULL. Returns Acad::eInvalidInput if the input name does not conform to Extended Symbol Name rules.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileGenerator Class

[Classes](#)

Class AcMapMbTileGenerator is a pure virtual base class which defines the interface for the Tile Generators.

```
class AcMapMbTileGenerator;  
File
```

AcMapMbTileGenerator.h

Remarks

All Tile Generators have the same interface and are closely linked with their respective Settings classes. The TileGenerator should not be instantiated directly, but must be obtained from the appropriate settings class. A settings class of the appropriate type must be passed back into the GenerateTiles function.

☐ Methods



[~AcMapMbTileGenerator](#) Destroys an instance of this class.



[AcMapMbTileGenerator](#) Constructs an instance of this class.

[GenerateTiles](#) Creates the tile set from the settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileGeneratorSettings Class

[Classes](#)

This is a list of values we handle at the level of the base class It is mostly for the creation and maintenance of the settings classes Map Book source Map Book Scale Scale DM Map name Tiling scheme DisableEmptyTiles IntersectEntites InteresectEntitySet OverlapPercent Class AcMapMbTileGeneratorSettings is the base settings class for tile generation.

```
class AcMapMbTileGeneratorSettings : public AcDbObject;  
File
```

AcMapMbTileGeneratorSettings.h

Remarks

It contains the properties needed by all derived classes. The user will not instantiate a member of this class as the derived classes have some properties needed by actual tiling schemes. The Tile Generators and the Tile Generator Settings classes have a direct relationship. For any future expansion, users will typically create both a new Generator and Settings class. The appropriate Generator will be obtained from the Settings class. The DECLARE\_MEMBERS macro is used in the declaration of classes that are to be a part of the ObjectARX run-time tree.

☐ Enumerations

 [EMapBookSource](#)

This enum defines the different sources for a MapBook.

 [ETilingScheme](#)

This enum defines the different tiling schemes we have available.

☐ Methods



[~AcMapMbTileGeneratorSettings](#)

Destroys an instance of this class.



[AcMapMbTileGeneratorSettings](#)

Constructs an instance of this class.

[ActiveTilingScheme](#)

Returns an enum which specifies the active tiling scheme.

## [AllocateTileGenerator](#)

This method is used to allocate a tile generator that can be used by the settings object.

## [copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

## [DisableEmptyTiles](#)

Returns the boolean value whether to disable empty tiles.

## [DMapName](#)

The pointer points to memory holding the dm map name. It is recommended that if you are going to keep this pointer for any length of time you make a copy of it as the contents can be changed at any time. The user should not attempt to delete this memory. This string can be empty to indicate model space but will never be NULL.

## [DmScaleFactor](#)

Returns the scale factor of the DM Map for the Map Book.

## [dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

## [dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

## [dxInFields](#)

Lets this object read its data. See also `dxInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

<a href="#"><u>dxfoutfields</u></a>	Lets this object write its data. See also dxfoutfields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.
<a href="#"><u>GetIntersectEntitySet</u></a>	Returns the selection set of entities intersected.
<a href="#"><u>Implementation</u></a>	Returns the implementation object.
<a href="#"><u>IntersectEntities</u></a>	Returns the boolean value whether to use a particular set of entities as an intersection set when determining which tiles are empty.
<a href="#"><u>MapBookSource</u></a>	Methods used to store and retrieve the data. Returns an enum which specifies the Source type.
<a href="#"><u>OverlapPercent</u></a>	Returns the Overlap percent.
<a href="#"><u>ScaleFactor</u></a>	Returns the scale factor for the Map Book.
<a href="#"><u>SetActiveTilingScheme</u></a>	Sets an enum which specifies the active tiling scheme.
<a href="#"><u>SetDisableEmptyTiles</u></a>	Sets the boolean value whether to disable empty tiles.
<a href="#"><u>SetDMMapName</u></a>	Sets the DM Map name to be used in the tiler.
<a href="#"><u>SetDmScaleFactor</u></a>	Sets the scale of the DM Map for the Map Book.
<a href="#"><u>SetIntersectEntities</u></a>	Sets the boolean value whether to use the intersection set.
<a href="#"><u>SetIntersectEntitySet</u></a>	Sets the selection set of entities for determining empty tiles.
<a href="#"><u>SetMapBookSource</u></a>	Sets an enum which specifies the Source type.
<a href="#"><u>SetOverlapPercent</u></a>	Sets the Overlap data member.
<a href="#"><u>SetScaleFactor</u></a>	Sets the scale data member.
<a href="#"><u>SetTitleLayerName</u></a>	Sets tthe tile layer name to be used in the tiler.
<a href="#"><u>TileLayerName</u></a>	Returns a pointer to memory holding the tile

layer name.

### [xmlInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [xmlOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetDMMMapName Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the DM Map name to be used in the tiler.

```
Acad::ErrorStatus SetDMMMapName(  
    const ACHAR* pszDMMMapName  
);
```

Parameters	Description
pszDMMMapName	Input Display Manager Map name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: SetTileLayerName Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets tthe tile layer name to be used in the tiler.

```
virtual Acad::ErrorStatus SetTileLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input layer name where tile reference polygons will be created.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: xmlInFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: xmlOutFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileGeneratorSettingsArea Class

[Classes](#)

Class AcMapMbTileGeneratorSettingsArea contains the added properties needed for area tile generation.

```
class AcMapMbTileGeneratorSettingsArea : public AcMapMbTileGenerator  
File
```

AcMapMbTileGeneratorSettingsArea.h

Remarks

These are the start point and the end point.

☐ Methods



[~AcMapMbTileGeneratorSettingsArea](#) Destroys an instance of this class.



[AcMapMbTileGeneratorSettingsArea](#) Constructs an instance of this class.

[AllocateTileGenerator](#)

This method is used to allocate a tile generator that can be used by the settings object.

[AreaFirstPoint](#)

Returns the first corner point of the area to tile.

[AreaLastPoint](#)

Return the second or opposite corner point of the area to tile.

[clone](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD

[copyFrom](#)

ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[GetAreaPoints](#)

Returns the extent points of the area to tile.

[GetExtents](#)

Gets the extents of the area to tile.

[SetAreaFirstPoint](#)

Sets the first corner point of the area to tile.

[SetAreaLastPoint](#)

Sets the second or opposite corner point of the area to tile.

[SetAreaPoints](#)

Sets the start point of the area to tile.

[SetExtents](#)

Sets the extents of the area to tile.

### [xmlInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [xmlOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: xmlInFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: xmlOutFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileGeneratorSettingsGrid Class

[Classes](#)

This is a list of values we handle pertinent to a Grid Tiler.

```
class AcMapMbTileGeneratorSettingsGrid : public AcMapMbTileGenerator  
File
```

AcMapMbTileGeneratorSettingsGrid.h

Remarks

Start point NumberOfRows NumberOfColumns Class

AcMapMbTileGeneratorSettingsGrid contains the added properties needed for grid tile generation. These are the start point, and the number of rows and columns.

☐ Methods



[~AcMapMbTileGeneratorSettingsGrid](#) Destroys an instance of this class.



[AcMapMbTileGeneratorSettingsGrid](#) Constructs an instance of this class.

[AllocateTileGenerator](#)

This method is used to allocate a tile generator that can be used by the settings object.

[AreaStartPoint](#)

Return the start point of the area to tile.

[clone](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the

system as needed; it is unlikely that you will need to call it directly.

### [dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfinFields](#)

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [NumberOfColumns](#)

Returns the integer value indicating number of columns.

### [NumberOfRows](#)

Returns the integer value indicating number of rows.

### [SetAreaStartPoint](#)

Set the start point of the area to tile.

### [SetNumberOfColumns](#)

Sets the integer value indicating number of columns.

### [SetNumberOfRows](#)

Sets the integer value indicating number of rows.

Lets this object read its data. See also `dwgInFields()` in the AutoCAD

## [xmlInFields](#)

ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

## [xmlOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: xmlInFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid::xmlOutFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileGeneratorSettingsManual Class

[Classes](#)

This is a list of values we handle pertinent to a Manual Tiler Selection set of plines Class AcMapMbTileGeneratorSettingsManual contains the added properties needed for manual tile generation.

```
class AcMapMbTileGeneratorSettingsManual : public AcMapMbTileGenerat  
File
```

AcMapMbTileGeneratorSettingsManual.h

Remarks

This is the selection set of plines.

☐ Methods



[~AcMapMbTileGeneratorSettingsManual](#) Destroys an instance of this class.



[AcMapMbTileGeneratorSettingsManual](#) Constructs an instance of this class.

[AllocateTileGenerator](#)

This method is used to allocate a tile generator that can be used by the settings object.

[clone](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[GetSelectedTileBoundaries](#)

Returns the object ids of of entities selected for tile boundaries.

[SetSelectedTileBoundaries](#)

Sets the selection set of entities selected.

[SetTitleLayerName](#)

Overriding base class version.

[TileLayerName](#)

This is overridden from the base class since layer name is not appropriate for manual selection.

[xmlInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the

system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[xmlOutFields](#)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: SetTileLayerName Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Overriding base class version.

```
virtual Acad::ErrorStatus SetTileLayerName(  
    const ACHAR* pszLayerName  
);
```

Parameters	Description
pszLayerName	Input Layer name.

Returns

Returns Acad::eNotApplicable.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: xmlInFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: xmlOutFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGenerator Class

[Classes](#)

Class AcMapMbTileNameGenerator is a pure virtual base class which defines the interface for the Name Generators.

```
class AcMapMbTileNameGenerator;
```

File

AcMapMBTileNameGenerator.h

Remarks

All Name Generators have the same interface and are closely linked with their respective Settings classes. The TileNameGenerator should not be instantiated directly, but must be obtained from the appropriate settings class. A settings class of the appropriate type must be passed back into the GenerateTileNames function.

☐ Methods



[~AcMapMbTileNameGenerator](#) Destroys an instance of this class.



[AcMapMbTileNameGenerator](#) Constructs an instance of this class.

[GenerateTileNames](#) Iterates through tile set and sets the names of individual tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorData Class

[Classes](#)

Class AcMapMbTileNameGeneratorData inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) which defines the interface for the Name Generators.

```
class AcMapMbTileNameGeneratorData : public AcMapMbTileNameGenerator  
File
```

AcMapMBTileNameGeneratorData.h

Remarks

All Name Generators have the same interface and are closely linked with their respective Settings classes. The TileNameGenerator should not be instantiated directly, but must be obtained from the appropriate settings class. A settings class of type [AcMapMbTileNameGeneratorSettingsData](#) must be passed back into the GenerateTileNames function.

☐ Methods

-  [~AcMapMbTileNameGeneratorData](#) Destroys an instance of this class.
-  [AcMapMbTileNameGeneratorData](#) Constructs an instance of this class.
- [GenerateTileNames](#) Iterates through tile set and sets the names of individual tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorGrid Class

[Classes](#)

Class AcMapMbTileNameGeneratorGrid inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) which defines the interface for the Name Generators.

```
class AcMapMbTileNameGeneratorGrid : public AcMapMbTileNameGenerator  
File
```

AcMapMBTileNameGeneratorGrid.h

Remarks

All Name Generators have the same interface and are closely linked with their respective Settings classes. The TileNameGenerator should not be instantiated directly, but must be obtained from the appropriate settings class. A settings class of type [AcMapMbTileNameGeneratorSettingsGrid](#) must be passed back into the GenerateTileNames function.

☐ Methods



[~AcMapMbTileNameGeneratorGrid](#) Destroys an instance of this class.



[AcMapMbTileNameGeneratorGrid](#) Constructs an instance of this class.

[GenerateTileNames](#)

Iterates through tile set and sets the names of individual tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorIndexer Class

[Classes](#)

This class encapsulates a naming nugget.

```
class AcMapMbTileNameGeneratorIndexer;  
File
```

AcMapMbTileNameGeneratorSettings.h

Remarks

Basically it stores whether it's alphabetic or numeric, what the start index is and what the increment is. The NameGeneratorSettings classes will supply either one or 2 of these. A sequential naming will need one, and a Row Column naming scheme will need both a Row and a Column indexer. The class is read only in that once you have created it, you can only look at its properties. It is saved and restored from the dwg or dxf file containing the map book it is associated with using the StreamIn, StreamOut functions. These are not typically called directly.

☐ Enumerations



This is record

[EIndexOrientation](#) AcMapMbTileNameGeneratorIndexer::EIndexOrientation.

☐ Methods



[~AcMapMbTileNameGeneratorIndexer](#) Destroys an instance of this class.



[AcMapMbTileNameGeneratorIndexer](#) Constructs an instance of this class.

[Increment](#)

Retrieves the increment value for the index.

[Index](#)

Retrieves the starting Index.

[IsInverted](#)

Retrieves whether the indexing should start at the top or front or at the back or bottom.

[IsNumeric](#)

Retrieves whether the indexing is numeric or alphabetic.

[Orientation](#)

Retrieves the orientation or type of Index.

[StreamIn](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

[StreamIn](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

[StreamIn](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

[StreamOut](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

[StreamOut](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

[StreamOut](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamIn Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamIn(  
    AcMapMbXmlFiler* pFiler  
);
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamOut Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamOut(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorSequence Class

[Classes](#)

Class AcMapMbTileNameGeneratorSequence inherits from the pure virtual base class [AcMapMbTileNameGenerator](#) which defines the interface for the Name Generators.

```
class AcMapMbTileNameGeneratorSequence : public AcMapMbTileNameGener  
File
```

AcMapMBTileNameGeneratorSequence.h

Remarks

All Name Generators have the same interface and are closely linked with their respective Settings classes. The TileNameGenerator should not be instantiated directly, but must be obtained from the appropriate settings class. A settings class of type [AcMapMbTileNameGeneratorSettingsSequence](#) must be passed back into the GenerateTileNames function.

☐ Methods

 [~AcMapMbTileNameGeneratorSequence](#) Destroys an instance of this class.

 [AcMapMbTileNameGeneratorSequence](#) Constructs an instance of this class.

[GenerateTileNames](#) Iterates through tile set and sets the names of individual tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorSettings Class

[Classes](#)

Class AcMapMbTileNameGeneratorSettings is the base settings class for name generation.

```
class AcMapMbTileNameGeneratorSettings : public AcDbObject;  
File
```

AcMapMbTileNameGeneratorSettings.h

Remarks

It contains the properties needed by all derived classes. The user will not instantiate a member of this class as the derived classes have some properties needed by actual naming schemes. The Name Generators and the Name Generator Settings classes have a direct relationship. For any future expansion, users will typically create both a new Generator and Settings class. The appropriate Generator will be obtained from the Settings class. The DECLARE\_MEMBERS macro is used in the declaration of classes that are to be a part of the ObjectARX run-time tree.

☐ Enumerations



[ETileNamingScheme](#) &nbsp;  

☐ Methods



[~AcMapMbTileNameGeneratorSettings](#) Destroys an instance of this class.



[AcMapMbTileNameGeneratorSettings](#) Constructs an instance of this class.

[AllocateTileNameGenerator](#)

Returns the associated Name Generator.

[copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the

[CountDisabledTiles](#)

system as needed; it is unlikely that you will need to call it directly.

Returns the count disable tiles data member.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[GetAlphaSequence](#)

&nbsp;

[Implementation](#)

Returns the implementation object.

[SetCountDisabledTiles](#)

Sets the count disable tiles data member.

[SetTileNamingScheme](#)

Returns an enum which specifies the tile naming scheme.

[TileNamingScheme](#)

Returns an enum which specifies the tile naming scheme.

Lets this object read its data. See also

### [xmlInFields](#)

dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [xmlOutFields](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: GetAlphaSequence Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

&nbsp;

```
static Acad::ErrorStatus GetAlphaSequence(  
    ACHAR*& pszAlphaSequence  
);
```

Parameters	Description
------------	-------------

pszAlphaSequence	
------------------	--

Returns

Returns the implementation, or NULL if unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: xmlInFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: xmlOutFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorSettingsData Class

[Classes](#)

Class AcMapMbTileNameGeneratorSettingsData contains the added property needed for data name generation.

```
class AcMapMbTileNameGeneratorSettingsData : public AcMapMbTileNameG  
File
```

AcMapMbTileNameGeneratorSettingsData.h

Remarks

This is the expression needed to get the tile name from the reference object for the tile. The reference objects will typically be rectangular polygons with object data or some other property supplying the tile name.

☐ Methods



[~AcMapMbTileNameGeneratorSettingsData](#) Destroys an instance of this class.



[AcMapMbTileNameGeneratorSettingsData](#) Constructs an instance of this class.

[AllocateTileNameGenerator](#) Returns the associated Name Generator.

[clone](#) Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#) Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This

overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfInFields](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [dxfOutFields](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [Expression](#)

Retrieves the expression to be applied to the reference object.

### [SetExpression](#)

Sets the expression to be applied to the reference object.

### [xmlInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

### [xmlOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: SetExpression Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Sets the expression to be applied to the reference object.

```
Acad::ErrorStatus SetExpression(  
    const ACHAR* pszExpression  
);
```

Parameters	Description
pszExpression	Variable to set

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: xmlInFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData::xmlOutFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorSettingsGrid Class

[Classes](#)

Class AcMapMbTileNameGeneratorSettingsGrid contains the added properties needed for grid name generation.

```
class AcMapMbTileNameGeneratorSettingsGrid : public AcMapMbTileNameG  
File
```

AcMapMbTileNameGeneratorSettingsGrid.h

Remarks

These are the primary index, the secondary index, and the separator string. The primary index could be either row or column, and the secondary index will be the other type. The primary index comes first in the tile name, followed by the separator string, followed by the secondary index.

☐ Enumerations

📄 [EGridNamingOrder](#) This enum indicates the naming order for the grid.

☐ Methods



[~AcMapMbTileNameGeneratorSettingsGrid](#) Destroys an instance of this class.



[AcMapMbTileNameGeneratorSettingsGrid](#) Constructs an instance of this class.

[AllocateTileNameGenerator](#)

Returns the associated Name Generator.

[clone](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Copies the contents of an object into the messaged object, if

[copyFrom](#)

feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgInFields](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxflnFields](#)

Lets this object read its data. See also `dxflnFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoOutFields](#)

Lets this object write its data. See also `dxfoOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call

[GetPrimaryIndex](#)

[GetSecondaryIndex](#)

[Separator](#)

[SetPrimaryIndex](#)

[SetSecondaryIndex](#)

[SetSeparator](#)

[xmlInFields](#)

[xmlOutFields](#)

it directly.

Returns the primary indexer.

Returns the column index data member.

Returns the separator data member.

Sets the primary indexer.

Sets the primary indexer.

Sets the separator data member.

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: SetSeparator Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Sets the separator data member.

```
Acad::ErrorStatus SetSeparator(  
    const ACHAR* pszSeparator  
);
```

Parameters	Description
sSeparator	Variable to hold value set

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: xmlInFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: xmlOutFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileNameGeneratorSettingsSequence Class

[Classes](#)

Class AcMapMbTileNameGeneratorSettingsSequence contains the added property needed for sequential name generation.

```
class AcMapMbTileNameGeneratorSettingsSequence : public AcMapMbTileNameGeneratorSettingsSequence  
File
```

AcMapMbTileNameGeneratorSettingsSequence.h

Remarks

This is the sequential index.

☐ Methods



[~AcMapMbTileNameGeneratorSettingsSequence](#)

Destroys an instance of this class.



[AcMapMbTileNameGeneratorSettingsSequence](#)

Constructs an instance of this class.

[AllocateTileNameGenerator](#)

Returns the associated Name Generator.

[clone](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[copyFrom](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden

[dwgInFields](#)

function is called by the system as needed; it is unlikely that you will need to call it directly.

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dwgOutFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfinFields](#)

Lets this object read its data. See also `dxfinFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[dxfoutFields](#)

Lets this object write its data. See also `dxfoutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

Retrieves the sequence

[GetIndex](#)

indexer.

[SetIndex](#)

Sets the primary indexer.

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[xmlInFields](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

[xmlOutFields](#)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: xmlInFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlInFields(  
    AcMapMbXmlFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence::xmlOutFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus xmlOutFields(  
    AcMapMbXmlFiler* pFiler  
) const;
```

Parameters

Description

`pFiler`

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMbTileSet Class

[Classes](#)

Provides functions to create, name, and return the tiles used by a MapBook.

```
class AcMapMbTileSet;
```

File

AcMapMbTileSet.h

Remarks

The UI can display the tiles in several ways. Numeric sequences are displayed differently from alphabetic sequences, and there may be differences between an Alpha sequence and an alphabetized sequence. If the tiles are returned as a sequence, then the root will contain a single child node, and this node will have several siblings. None of the siblings will have any children. If the requested return is RowCol, then.

☐ Enumerations

📄 [EMbNameSchema](#) This is record AcMapMbTileSet::EMbNameSchema.

📄 [EMbTreeType](#) This is record AcMapMbTileSet::EMbTreeType.

☐ Classes

[AcMapContainer](#) Alternate interface for tree.

[AcMapContainerIterator](#) This is class  
AcMapMbTileSet::AcMapContainerIterator.

[AcMapTileIterator](#) This is class AcMapMbTileSet::AcMapTileIterator.

[AcMapTreeNode](#) Class AcMapTreeNode models the most general  
structure for an organization of tiles.

☐ Methods

🔷 [~AcMapMbTileSet](#) Destroys an instance of this class.

🔷 [AcMapMbTileSet](#) Constructs an instance of this class.

[AddTiles](#) Sets the new tiles into the tile set.

[BuildAdjacentInfo](#) Generate adjacency information and assigns it to every  
tile.

<a href="#"><u>ColumnNumber</u></a>	Returns the number of columns in a grid tiling.
<a href="#"><u>ConstContainer</u></a>	Returns the root of the tree organized according to the tree type as a container.
<a href="#"><u>ConstTree</u></a>	Returns the root of the tree organized according to the tree type.
<a href="#"><u>CreateGridLayer</u></a>	Sets the layer on which the grid polygons will be created.
<a href="#"><u>CreateNode</u></a>	Creates a new node for the tree.
<a href="#"><u>Database</u></a>	Returns the pointer to the drawing database the tile set is associated with.
<a href="#"><u>Erase</u></a>	Removes the specified tile from the tile set.
<a href="#"><u>Find</u></a>	Returns the tile corresponding to the given tile id if it exists.
<a href="#"><u>GetExtents</u></a>	Returns the total extents of the view ports of this tile set.
<a href="#"><u>Implementation</u></a>	Returns the implementation object.
<a href="#"><u>LayerId</u></a>	When a tile set gets generated (except manual), it creates polylines per every tile and puts them to the specified layer.
<a href="#"><u>MapBook</u></a>	Returns the Id of the AcMapMbMapBook object it belongs to.
<a href="#"><u>NamingScheme</u></a>	Returns the way or schema used when naming the tiles.
<a href="#"><u>RowNumber</u></a>	Returns the number of rows in a grid tiling.
<a href="#"><u>SetAdjacentProperties</u></a>	Iterates a Tile Set and sets adjacent properties for the corresponding Sheets.
<a href="#"><u>StreamIn</u></a>	Reads data from a dwg file.
<a href="#"><u>StreamIn</u></a>	Reads data from a dxf file.
<a href="#"><u>StreamOut</u></a>	Loads data into a dwg file.
<a href="#"><u>StreamOut</u></a>	Loads data into a dxf file.
<a href="#"><u>TileNumber</u></a>	Returns the total number of tiles in the set.
<a href="#"><u>TilingScheme</u></a>	Returns the schema or organization used when creating the tiles.
<a href="#"><u>Tree</u></a>	Returns the root of the tree organized according to the tree type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet::CreateGridLayer Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Sets the layer on which the grid polygons will be created.

```
Acad::ErrorStatus CreateGridLayer(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input name of the layer to create polygons on. This must be a valid extended symbol name.
pDb	Input AcDbDatabase where the polygons will be drawn.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The database is needed to indicate where to create them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Map Book Global-Function API (Deprecated)

The global-function API for plotting map books is deprecated in favor of the [class-based API](#).

<a href="#">Map Book Function Catalog</a>	Map Book functions sorted by name.
<a href="#">Map Book Function Synopsis</a>	Map Book functions sorted by functional group.



# Map Book Function Catalog

[Map Book Global-Function API](#)

---

Map Book functions sorted by name.

**Note** The global-function API for plotting map books is deprecated in favor of the [class-based API](#).

[dwg](#) | [plt](#) | [topo](#)

[map\\_dwgbreakobj](#) [map\\_dwgtrimobj](#)

[map\\_pltblkatts](#)

[map\\_pltblklist](#)

[map\\_pltblkvps](#)

[map\\_pltcleanup](#)

[map\\_pltcurrdef](#)

[map\\_pltcurrdel](#)

[map\\_pltcurrget](#)

[map\\_pltcurrsave](#)

[map\\_pltcurrset](#)

[map\\_pltdefdelete](#)

[map\\_pltdefget](#)

[map\\_pltdeflist](#)

[map\\_pltdefread](#)

[map\\_pltdefsave](#)

[map\\_pltdefvalid](#)

[map\\_pltdefverify](#)

[map\\_pltdisplay](#)

[map\\_pltexecute](#)

[map\\_pltinit](#)

[map\\_pltplot](#)

[map\\_pltrestore](#)

[map\\_topoaudit](#)

[map\\_topoclose](#)

[map\\_topocomplete](#)

[map\\_topostat](#)



## Map Book Function Synopsis

[Map Book Global-Function API](#)

---

Map Book functions sorted by functional group.

**Note** The global-function API for plotting map books is deprecated in favor of the [class-based API](#).

[Boundary Functions](#) [Plotting Functions](#)

[Topology Functions](#)

## Object Filter

### ▣ Classes

Filters objects (entities) in the current drawing based on layer, feature-class, and block criteria.

Filters are constructed with "\*" as the default filtering value, indicating that all entities are to be filtered.

Other valid values are "" (an empty string) to ignore entities during filtering, and NULL (a null filter) to exclude any entities returned by the filter set. The

#### [AcDbBasicFilter](#)

default AcDbBasicFilter object is constructed with \* as the value for the Layers, FeatureClasses, and Blocks filtering criteria, and so will filter out classified blocks from all layers. The following table shows how to set entity-filtering criteria for various filtering scenarios. (Note that these scenarios do not include objects on frozen, locked, or off layers unless you have set the layer status mask with SetLayerStatusMask()).

#### [AcDbObjectFilter](#)

Base class for creating filters that filter objects (entities) in the current drawing based on the specified criteria.

#### [AcDbObjectFilterGroup](#)

Filters objects (entities) in the current drawing based on the criteria of one or more listed filters.

Links

[Object Filter](#)

Classes

[Object Filter](#)

☐ Classes

[AcDbBasicFilter](#)

Filters objects (entities) in the current drawing based on layer, feature-class, and block criteria. Filters are constructed with "\*" as the default filtering value, indicating that all entities are to be filtered. Other valid values are "" (an empty string) to ignore entities during filtering, and NULL (a null filter) to exclude any entities returned by the filter set. The default AcDbBasicFilter object is constructed with \* as the value for the Layers, FeatureClasses, and Blocks filtering criteria, and so will filter out classified blocks from all layers. The following table shows how to set entity-filtering criteria for various filtering... [more](#)

[AcDbObjectFilter](#)

Base class for creating filters that filter objects (entities) in the current drawing based on the specified criteria. Use this class to derive custom filters if you want to filter objects on criteria other than layers, feature classes, and blocks; otherwise, use the AcDbBasicFilterclass. Any class that you derive from this class must implement FilterObjects()to define the filtering criteria. To use multiple filters, use the [AcDbObjectFilterGroup](#)class.

[AcDbObjectFilterGroup](#)

Filters objects (entities) in the current drawing based on the criteria of one or more listed filters. To create a filter, construct an instance of the AcDbBasicFilterclass or an instance of a class derived from the AcDbObjectFilterclass.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcDbBasicFilter Class

[Classes](#)

Filters objects (entities) in the current drawing based on layer, feature-class, and block criteria.

Filters are constructed with "\*" as the default filtering value, indicating that all entities are to be filtered. Other valid values are "" (an empty string) to ignore entities during filtering, and NULL (a null filter) to exclude any entities returned by the filter set. The default AcDbBasicFilter object is constructed with \* as the value for the Layers, FeatureClasses, and Blocks filtering criteria, and so will filter out classified blocks from all layers. The following table shows how to set entity-filtering criteria for various filtering scenarios. (Note that these scenarios do not include objects on frozen, locked, or off layers unless you have set the layer status mask with SetLayerStatusMask()).

	Layers	FeatureClasses	Blocks
To select			
All objects from all layers	*	""	""
All classified objects from all layers	*	*	""
All blocks (inserts) from all layers	*	""	*
Classified blocks from all layers	*	*	*
Classified objects from specified layers	Layer1,Layer5	*	""
Blocks from specified layers	Layer1,Layer5	""	*
Specified classified objects from all layers	*	Road,Lake	""
Specified blocks from all layers	*	""	Block1,Block5
Specified classified objects			

and specified blocks from all layers	*	Road,Lake	Block1,Block5
Specified classified objects from specified layers	Layer1,Layer5	Road,Lake	""
Specified blocks from specified layers	Layer1,Layer5	""	Block1,Block5
Specified classified objects and specified blocks from specified layers	Layer1,Layer5	Road,Lake	Block1,Block5
All objects from all layers except classified objects	*	NULL	""
All objects from all layers except blocks	*	""	NULL
All objects from all layers except classified objects and blocks	*	NULL	NULL
All objects from specified layers except classified objects	Layer1,Layer5	NULL	""
All objects from specified layers except blocks	Layer1,Layer5	""	NULL

To use multiple filters, see the class [AcDbObjectFilterGroup](#).

```
class AcDbBasicFilter : public AcDbObjectFilter;
File
```

AcDbObjectFilter.h

#### ☐ Enumerations

 [ELayerStatus](#)

Enumerates the types of layer-status filters.

#### ☐ Methods

 [~AcDbBasicFilter](#)

Destroys an instance of this class.

⇒ [AcDbBasicFilter](#)

Constructs an instance of this class with the default value filter values set to "\*" to filter all entities.

⇒ [AcDbBasicFilter](#)

Constructs an instance of this class by using the specified layer, feature-class, and block filtering criteria.

[AddBlocks](#)

Adds a block to the list of blocks on which to filter objects.

[AddFeatureClasses](#)

Adds a feature class to the list of feature classes on which to filter objects.

[AddLayers](#)

Adds a layer to the list of layers on which to filter objects.

[Blocks](#)

Retrieves the block(s) on which to filter. The block filter is defined with SetBlocks() (twoforms).

[FeatureClasses](#)

Retrieves the feature class(es) on which to filter. The feature-class filter is defined with SetFeatureClasses() (twoforms).

Filters objects on the current drawing based on the layer, feature-class, and block criteria.

[FilterObjects](#)

A filtering criterion can take the following values: "\*" - (Default) Filters all entities that reside on layers with normal layer status. Unless SetLayerStatusMask() has been called with specific layer mask settings, all objects on layers which are not frozen, off, or locked are considered in the filtering process. "" (empty string) - Indicates that a particular filter is to be ignored during the filtering process. NULL - Indicates that particular filtered entities are to be excluded from the filtered set.

A name - The name of... [more](#)

[Layers](#)

Retrieves the layer(s) on which to filter. The layer filter is defined with SetLayers() (twoforms). NULL and empty strings ("") are invalid values.

[LayerStatusMask](#)

Determines whether the layer status mask has been set by using SetLayerStatusMask(). A nonzero return value indicates that objects on layers which are frozen, locked, or off are to be considered in the filtering process. A zero return value indicates that no layer mask is set and filtering will be applied to objects with

	normal layer status.
<a href="#">ResetBlocks</a>	Clears the filter list of blocks.
<a href="#">ResetFeatureClasses</a>	Clears the filter list of feature classes.
<a href="#">ResetLayers</a>	Clears the list of filter layers.
<a href="#">SetBlocks</a>	Sets the block filters. See also <a href="#">Blocks()</a> .
<a href="#">SetBlocks</a>	Sets the blocks on which to filter objects. See also <a href="#">Blocks()</a> .
<a href="#">SetFeatureClasses</a>	
<a href="#">SetFeatureClasses</a>	Sets the feature classes on which to filter objects. See also <a href="#">FeatureClasses()</a> .
<a href="#">SetLayers</a>	Sets the layers on which to filter objects. NULL and empty strings ("") are invalid values. See also <a href="#">Layers()</a> .
<a href="#">SetLayers</a>	Sets the layers on which to filter objects. See also <a href="#">Layers()</a> .
<a href="#">SetLayerStatusMask</a>	Sets the layer-status filter criterion that specifies whether objects on frozen, locked, or off layers are filtered. Use <a href="#">LayerStatusMask()</a> to determine whether the layer-status filter is set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter:: AcDbBasicFilter Constructor

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Constructs an instance of this class by using the specified layer, feature-class, and block filtering criteria.

```
AcDbBasicFilter(  
    const ACHAR* layerFilter,  
    const ACHAR* featureClassFilter,  
    const ACHAR* blockfilter  
);
```

Parameters	Description
layerFilter	Input layer name(s). Separate multiple names by commas. Layer names are case-insensitive.
featureClassFilter	Input feature-class name(s). Separate multiple names by commas.
blockfilter	Input block names. Separate multiple names by commas.

## Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: AddBlocks Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Adds a block to the list of blocks on which to filter objects.

```
Acad::ErrorStatus AddBlocks(  
    const ACHAR* blockFilter  
);
```

Parameters	Description
blockFilter	Input name of the block to add to the list.

Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)  
AcDbBasicFilter:: AddFeatureClasses Method  
[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Adds a feature class to the list of feature classes on which to filter objects.

```
Acad::ErrorStatus AddFeatureClasses(  
    const ACHAR* featureClassFilter  
);
```

Parameters	Description
featureClassFilter	Input name of the feature class to add to the list.

Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: AddLayers Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Adds a layer to the list of layers on which to filter objects.

```
Acad::ErrorStatus AddLayers(  
    const ACHAR* layerFilter  
);
```

Parameters	Description
layerFilter	Input name of the layer to add to the list. Layer names are case-insensitive for filtering operations but are displayed in their original case.

## Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: SetBlocks Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the block filters. See also Blocks().

```
Acad::ErrorStatus SetBlocks(  
    AcArray<ACHAR*>& blockFilters  
);
```

Parameters	Description
blockFilters	Input array of block names.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: SetBlocks Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the blocks on which to filter objects. See also [Blocks\(\)](#).

```
Acad::ErrorStatus SetBlocks(  
    const ACHAR* blockFilter  
);
```

Parameters	Description
blockFilter	Input block name(s) in any of the following forms: A single block name ("Block1", for example)A list of comma-separated block names ("Block1,Block2", for example)A "*" to filter all blocks (this is the default setting)A "" (empty string) to ignore blocks during filteringA NULL to exclude blocks from the filtered setBlock names with wildcard characters ("Block?" or "Block-*", for example)

## Returns

Returns `Acad::ErrorStatus eOk` if successful. Returns `Acad::ErrorStatus eInvalidInput` if parameter validation failed. Returns `Acad::ErrorStatus eInvalidInput` if the block list cannot be obtained (applies only if `blockFilter` is "\*")

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter:: SetFeatureClasses Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

```
Acad::ErrorStatus SetFeatureClasses(  
    const ACHAR* featureClassFilter  
);
```

Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed. Returns Acad::ErrorStatus eInvalidInput if the feature-class list cannot be obtained (applies only if featureClassFilter is "\*")

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter:: SetFeatureClasses Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the feature classes on which to filter objects. See also FeatureClasses().

```
Acad::ErrorStatus SetFeatureClasses(  
    const AcArray<ACHAR*>& featureClassFilters  
);
```

Parameters	Description
featureClassFilters	Input array of feature-class names.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: SetLayers Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the layers on which to filter objects. NULL and empty strings ("" ) are invalid values. See also Layers().

```
Acad::ErrorStatus SetLayers(  
    const ACHAR* layerFilter  
);
```

Parameters	Description
layerFilter	Input layer name(s). Separate multiple names by commas. If the layer filter is "*", and both block and feature-class filters are "" (ignore during filtering), then objects on all layers are filtered. Layer names are case-insensitive.

## Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: SetLayers Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the layers on which to filter objects. See also Layers().

```
Acad::ErrorStatus SetLayers(  
    const AcArray<ACHAR*>& layerFilters  
);
```

Parameters	Description
layerFilters	Input array of layer names. Layer names are case-insensitive.

## Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcDbObjectFilter Class

[Classes](#)

Base class for creating filters that filter objects (entities) in the current drawing based on the specified criteria. Use this class to derive custom filters if you want to filter objects on criteria other than layers, feature classes, and blocks; otherwise, use the `AcDbBasicFilter` class. Any class that you derive from this class must implement `FilterObjects()` to define the filtering criteria. To use multiple filters, use the [AcDbObjectFilterGroup](#) class.

```
class AcDbObjectFilter;
```

File

AcDbObjectFilter.h

☐ Methods

◆ [~AcDbObjectFilter](#) Destroys an instance of this class.

◆ [AcDbObjectFilter](#) Constructs an instance of this class.

[FilterObjects](#) Filters objects on the current drawing based on the filtering criteria.

[IsActive](#) Determines whether this filter is active.

Activates or deactivates this filter. When this filter is active, objects are filtered based on the filtering criteria when `FilterObjects()` in the derived class is called.

[SetActive](#) When this filter is inactive, no objects are filtered. This setting applies only when the [AcDbObjectFilterGroup](#) class is used to filter objects.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcDbObjectFilterGroup Class

[Classes](#)

Filters objects (entities) in the current drawing based on the criteria of one or more listed filters. To create a filter, construct an instance of the AcDbBasicFilterclass or an instance of a class derived from the AcDbObjectFilterclass.

```
class AcDbObjectFilterGroup : public AcDbObjectFilter;  
File
```

AcDbObjectFilter.h

☐ Methods



[~AcDbObjectFilterGroup](#) Destroys an instance of this class.



[AcDbObjectFilterGroup](#) Constructs an instance of this class.

[AddObjectFilter](#) Adds a filter to the end of the filter list.

[FilterObjects](#) Filters objects on the current drawing based on the filtering criteria of all the filters in the filter list.

[GetObjectFilter](#) Retrieves a filter from the filter list.

[InsertObjectFilter](#) Inserts a filter into the filter list.

[IsEmpty](#) Determines whether the filter list is empty.

[ObjectFilterCount](#) Counts the number of filters in the filter list.

[RemoveAllObjectFilter](#) Clears the filter list.

[RemoveObjectFilter](#) Removes a filter from the filter list.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

# Oracle Spatial Advanced

## Functions

<a href="#">AcMapOSEGetConnection</a>	Gets the connection object.
---------------------------------------	-----------------------------

## Classes

<a href="#">AcMapOSEConnection</a>	
<a href="#">AcMapOSEConnectionReactor</a>	
<a href="#">AcMapOSEExport</a>	
<a href="#">AcMapOSEExportReactor</a>	
<a href="#">AcMapOSEImport</a>	
<a href="#">AcMapOSEImportReactor</a>	
<a href="#">AcMapOSEObject</a>	
<a href="#">AcMapOSEProject</a>	
<a href="#">AcMapOSEQuery</a>	

## Oracle Spatial (Deprecated)

### Functions

<a href="#">AcMapOracleGetConnection</a>	Gets the connection object.
--	-----------------------------

### Classes

<a href="#">AcMapOracleConnection</a>	
<a href="#">AcMapOracleConnectionReactor</a>	
<a href="#">AcMapOracleExport</a>	
<a href="#">AcMapOracleExportReactor</a>	
<a href="#">AcMapOracleIdentification</a>	
<a href="#">AcMapOracleImport</a>	
<a href="#">AcMapOracleImportReactor</a>	
<a href="#">AcMapOracleQuery</a>	

## Links

[Structs, Records, Enums](#)

Topology: Class-Based API

[Structs, Records, Enums](#)

▫ Namespaces

[AcMapTopologyManager](#) The topology manager object.

▫ Types

[AcMapFullEdgePtrArray](#) Dynamic array of

[AcMapHalfEdgePtrArray](#) Dynamic array of

[AcMapNodePtrArray](#) Dynamic array of

[AcMapObjectPtrArray](#) Dynamic array of

[AcMapOverlayDataArray](#) This is type AcMapOverlayDataArray.

[AcMapPolygonPtrArray](#) Dynamic array of

[AcMapRingPtrArray](#) Dynamic array of

[AcMapTopologySourceArray](#) This is type AcMapTopologySourceArray.

▫ Classes

[AcMapEntityCreationSettings](#) Settings for creating entities.

[AcMapFloodParameters](#) Settings for flood parameters.

[AcMapMarkerStyles](#) Settings for error markers.

[AcMapNetAnalysisParameters](#) Settings for analysis parameters.

[AcMapPointCreationSettings](#) Settings for creating points.

[AcMapTopoElement](#) Manages topology objects.

[AcMapTopoElementPtrArray](#) Template for dynamic arrays of topology element pointers.

[AcMapTopoFullEdge](#) Manages topology full edges.

[AcMapTopoHalfEdge](#) Manages topology half edges.

[AcMapTopoIterator](#) An iterator over a collection of

[AcMapTopology](#) The main topology object.

[AcMapTopologySource](#) Stores the topology source information. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoNode](#) Manages topology nodes.

[AcMapTopoOverlayData](#) For topology overlay analysis, use this class to set the data attributes that contain object data from linked database to include in the resulting

topology, and the name and description of the new object data table.

[AcMapTopoPolygon](#)

Manages topology polygons.

[AcMapTopoRing](#)

Manages topology rings.

[AcMapTraceParameters](#)

Settings for trace parameters.

☐ Structs, Records, Enums

[AcMapObjectDataField](#)

Information about a field in a topology object data table.

[AcMapObjectDataTable](#)

Information about a topology object data table.

[ETopologyMarkType](#)

Topology error-marker shapes.

[ETopologyType](#)

Topology types.

☐ Links

[Structs, Records, Enums](#)

Links

[Topology: Class-Based API](#)

AcMapTopologyManager Namespace

[Topology: Class-Based API](#)

The topology manager object. For more information, search for *topology* in AutoCAD Map Help.

☐ Enumerations

📄 [ETopologyScope](#) Enumerates the scope of a topology.

☐ Functions

[Delete](#)

Deletes the specified topology from the current drawing and all source drawings.

[GetTopologyScope](#)

Retrieves the scope of the specified topology. The returned value is cast to an ETopologyScopevalue to indicate whether the specified topology exists on the current drawing, source drawing, or both.

[GetTopologySource](#)

Retrieves the source information of the specified topology.

[Rename](#)

Renames the existing topology.

[TopologyExists](#)

Determines whether a specified topology exists in current or source (attached) drawing.

[TopologyExists](#)

Determines whether a specified topology exists in specified source.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager:: ETopologyScope Enumeration

[AcMapTopologyManager Namespace](#)

Enumerates the scope of a topology.

```
enum ETopologyScope {  
    eCurrentDwg = 0x0001,  
    eSourceDwg = 0x0002  
};  
File
```

AcMapTopologyManager.h

Parameters	Description
eCurrentDwg	The topology is in the current drawing.
eSourceDwg	The topology is in the source (attached) drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager:: Delete Function

[AcMapTopologyManager Namespace](#)

Deletes the specified topology from the current drawing and all source drawings.

```
AcMap::EErrorCode Delete(  
    const ACHAR* pszName,  
    bool bDelEntities  
);  
File
```

AcMapTopologyManager.h

Parameters	Description
pszName	Input name of the topology to delete.
bDelEntities	Input true to delete all the entities; otherwise, false.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Remarks

You must close a topology before you delete it.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager:: GetTopologyScope Function

[AcMapTopologyManager Namespace](#)

Retrieves the scope of the specified topology. The returned value is cast to an ETopologyScopevalue to indicate whether the specified topology exists on the current drawing, source drawing, or both.

```
int GetTopologyScope(  
    const ACHAR* pszName  
);
```

File

AcMapTopologyManager.h

Parameters	Description
pszName	Input name of a the topology to check.

Returns

Returns eCurrentDwg if the topology exists on current drawing. Returns eSourceDwg if the topology exists on source drawing. Returns eCurrentDwg | eSourceDwg if the topology exists on both current and source drawings. Returns 0 if the specified topology does not exist.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager:: GetTopologySource Function

[AcMapTopologyManager Namespace](#)

Retrieves the source information of the specified topology.

```
AcMap::EErrorCode GetTopologySource(  
    AcMapTopologySourceArray& apSource,  
    const ACHAR* pszName  
);  
File
```

AcMapTopologyManager.h

Parameters	Description
apSource	Output array of topology source information.
pszName	Input name of the topology.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager::Rename Function

[AcMapTopologyManager Namespace](#)

Renames the existing topology.

```
AcMap::EErrorCode Rename(  
    const ACHAR* pszOldName,  
    const ACHAR* pszNewName  
);  
File
```

AcMapTopologyManager.h

Parameters	Description
pszOldName	Input the existing name of the topology model.
pszNewName	Input the new name of the topology model.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager::TopologyExists Function

[AcMapTopologyManager Namespace](#)

Determines whether a specified topology exists in current or source (attached) drawing.

```
bool TopologyExists(  
    const ACHAR* pszName  
);  
File
```

AcMapTopologyManager.h

Parameters	Description
pszName	Input name of the topology to check.
Returns	

Returns true if the topology exists; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologyManager Namespace](#)

AcMapTopologyManager::TopologyExists Function

[AcMapTopologyManager Namespace](#)

Determines whether a specified topology exists in specified source.

```
bool TopologyExists(  
    const ACHAR* pszName,  
    ETopologyScope enmSource  
);  
File
```

AcMapTopologyManager.h

Parameters	Description
pszName	Input name of the topology to check.
enmSource	Input ETopologyScopetopology source.

Returns

Returns true if the topology exists in specified source; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Topology: Class-Based API](#)

Classes

[Topology: Class-Based API](#)

▣ Classes

[AcMapEntityCreationSettings](#)

Settings for creating entities. For more information, search for *topology* in AutoCAD Map Help.

[AcMapFloodParameters](#)

Settings for flood parameters. For more information, search for *topology* in AutoCAD Map Help.

[AcMapMarkerStyles](#)

Settings for error markers. For more information, search for *topology* in AutoCAD Map Help.

[AcMapNetAnalysisParameters](#)

Settings for analysis parameters. For more information, search for *topology* in AutoCAD Map Help.

[AcMapPointCreationSettings](#)

Settings for creating points. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoElement](#)

Manages topology objects. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoFullEdge](#)

Manages topology full edges. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoHalfEdge](#)

Manages topology half edges. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoIterator](#)

An iterator over a collection of AcMapTopologyinstances. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopology](#)

The main topology object. For more information, search for *topology* in AutoCAD Map Help.

Stores the topology source information. For

[AcMapTopologySource](#)

more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoNode](#)

Manages topology nodes. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoOverlayData](#)

For topology overlay analysis, use this class to set the data attributes that contain object data from linked database to include in the resulting topology, and the name and description of the new object data table. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoPolygon](#)

Manages topology polygons. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTopoRing](#)

Manages topology rings. For more information, search for *topology* in AutoCAD Map Help.

[AcMapTraceParameters](#)

Settings for trace parameters. For more information, search for *topology* in AutoCAD Map Help.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapEntityCreationSettings Class

[Classes](#)

Settings for creating entities. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapEntityCreationSettings;
```

File

AcMapTopology.h

☐ Methods



[~AcMapEntityCreationSettings](#) Destroys an instance of this class.



[AcMapEntityCreationSettings](#) Constructs an instance of this class.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapEntityCreationSettings Class](#), [AcMapEntityCreationSettings Class](#)  
AcMapEntityCreationSettings:: AcMapEntityCreationSettings Constructor  
[AcMapEntityCreationSettings Class](#) | [AcMapEntityCreationSettings Class](#)

Constructs an instance of this class.

```
AcMapEntityCreationSettings(  
    const ACHAR* pszLayerName,  
    int nColor  
);
```

Parameters	Description
pszLayerName	Input name of the layer for the created entity.
nColor	Input color index.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapFloodParameters Class

[Classes](#)

Settings for flood parameters. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapFloodParameters : public AcMapNetAnalysisParameters;  
File
```

AcMapTopology.h

☐ Methods



[~AcMapFloodParameters](#) Destroys an instance of this class.



[AcMapFloodParameters](#) Constructs an instance of this class.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapFloodParameters Class](#), [AcMapFloodParameters Class](#)

[AcMapFloodParameters::AcMapFloodParameters Constructor](#)

[AcMapFloodParameters Class](#) | [AcMapFloodParameters Class](#)

Constructs an instance of this class.

```
AcMapFloodParameters(  
    double dMaxResistance,  
    const ACHAR* aNodeResistanceExpression,  
    const ACHAR* aLinkDirectionExpression,  
    const ACHAR* aLinkResistanceExpression,  
    const ACHAR* aLinkReverseResistanceExpression,  
    bool bUseReverseDirection,  
    const ACHAR* aName = NULL,  
    const ACHAR* aDescription = NULL  
);
```

Parameters	Description
dMaxResistance	Input maximum resistance.
aNodeResistanceExpression	Input node resistance expression.
aLinkDirectionExpression	Input link direction expression.
aLinkResistanceExpression	Input link resistance expression.
aLinkReverseResistanceExpression	Input link reverse resistance expression.
bUseReverseDirection	Input true to use reverse direction; otherwise, false.
aName	Input name of the new topology. If this value is NULL, no new topology is created. The default value is NULL.
aDescription	Input description of the new topology. The default value is NULL.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapMarkerStyles Class

[Classes](#)

Settings for error markers. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapMarkerStyles;
```

File

AcMapTopology.h

☐ Methods

◆ [~AcMapMarkerStyles](#)

Destroys an instance of this class.

[SetMarkerSize](#)

Sets the error-marker size.

[SetTopoDuplicateCentroidMarker](#)

Sets the marker shape and color for a duplicate-centroid error.

[SetTopoIncompleteAreaMarker](#)

Sets the marker shape and color for an incomplete-area error.

[SetTopoIntersectionMarker](#)

Sets the marker shape and color for an intersection error.

[SetTopoMissingCentroidMarker](#)

Sets the marker shape and color for a missing-centroid error.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapNetAnalysisParameters Class

[Classes](#)

Settings for analysis parameters. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapNetAnalysisParameters;
```

File

AcMapTopology.h

☐ Methods



[~AcMapNetAnalysisParameters](#) Destroys an instance of this class.



[AcMapNetAnalysisParameters](#) Constructs an instance of this class.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapNetAnalysisParameters Class](#), [AcMapNetAnalysisParameters Class](#)  
AcMapNetAnalysisParameters:: AcMapNetAnalysisParameters Constructor  
[AcMapNetAnalysisParameters Class](#) | [AcMapNetAnalysisParameters Class](#)

Constructs an instance of this class.

```
AcMapNetAnalysisParameters(  
    double dMinResistance,  
    double dMaxResistance,  
    const ACHAR* aNodeResistanceExpression,  
    const ACHAR* aLinkDirectionExpression,  
    const ACHAR* aLinkResistanceExpression,  
    const ACHAR* aLinkReverseResistanceExpression,  
    bool bUseReverseDirection,  
    const ACHAR* aName = NULL,  
    const ACHAR* aDescription = NULL  
);
```

Parameters	Description
dMinResistance	Input minimum resistance.
dMaxResistance	Input maximum resistance.
aNodeResistanceExpression	Input node resistance expression.
aLinkDirectionExpression	Input link direction expression.
aLinkResistanceExpression	Input link resistance expression.
aLinkReverseResistanceExpression	Input link reverse resistance expression.
bUseReverseDirection	Input true to use reverse direction; otherwise, false.
aName	Input name of the new topology. The default value is NULL.
aDescription	Input description of the new topology. The default value is NULL.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapPointCreationSettings Class

[Classes](#)

Settings for creating points. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapPointCreationSettings : public AcMapEntityCreationSetting  
File
```

AcMapTopology.h

☐ Methods



[~AcMapPointCreationSettings](#) Destroys an instance of this class.



[AcMapPointCreationSettings](#) Constructs an instance of this class.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapPointCreationSettings Class](#), [AcMapPointCreationSettings Class](#)  
AcMapPointCreationSettings:: AcMapPointCreationSettings Constructor  
[AcMapPointCreationSettings Class](#) | [AcMapPointCreationSettings Class](#)

Constructs an instance of this class.

```
AcMapPointCreationSettings(  
    const ACHAR* pszLayerName,  
    int nColor,  
    bool bCreate,  
    const ACHAR* pszBlockname  
);
```

Parameters	Description
pszLayerName	Input name of layer name for the created node.
nColor	Input color index.
bCreate	Input true if creating a node; otherwise, false.
pszBlockname	Input name of the block for the created node.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoElement Class

[Classes](#)

Manages topology objects. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoElement;
```

File

AcMapTopoObject.h

☐ Methods



[~AcMapTopoElement](#) Destroys an instance of this class.



[AcMapTopoElement](#) Constructs an instance of this class.

[GetEntity](#) Retrieves the object identifier of the AutoCAD entity associated with this object.

[GetID](#) Retrieves the unique identifier of this object in its topology.

[IsOnThisObject](#) Determines whether a specified point is within tolerance of this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoElementPtrArray Template

[Classes](#)

Template for dynamic arrays of topology element pointers. Memory allocated for such arrays is released automatically when they go out of scope.

Most of the member functions of this class are defined in its parent class, AcArray.

```
template <class T>  
class AcMapTopoElementPtrArray : public AcArray<T *>;  
File
```

AcMapTopologyCommonDef.h

☐ Methods



[~AcMapTopoElementPtrArray](#) Destroys the array.

[Empty](#) Releases memory allocated to the array.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoFullEdge Class

[Classes](#)

Manages topology full edges. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoFullEdge : public AcMapTopoElement;
```

File

AcMapTopoFullEdge.h

☐ Enumerations



[EDirection](#)

Enumerates the directions of a full edge.

☐ Methods



[~AcMapTopoFullEdge](#)

Destroys an instance of this class.

[GetDirection](#)

Retrieves the direction of this edge.

[GetEntity](#)

Retrieves the object identifier of the AutoCAD entity associated with this edge.

[GetHalfEdge](#)

Retrieves the half edge that traverses one way along this full edge.

[GetID](#)

Retrieves the identifier of this edge in its topology.

[GetLength](#)

Retrieves the length of this edge.

[GetNextEdge](#)

Retrieves the next edge by moving forward or backward along this edge and turning left or right at the next node.

[GetNextNode](#)

Retrieves the next node by moving forward or backward along this edge.

[GetPolygon](#)

Retrieves the polygon associated with this edge (default level).

[GetResistance](#)

Retrieves the resistance of this edge.

[GetRing](#)

Retrieves the ring associated with this edge (default level).

[GetTopology](#)

Retrieves the topology that contains this edge.

[IsOnThisObject](#) Determines whether a specified point is within tolerance of this edge.

[SetDirection](#) Sets the direction of this edge.

[SetResistance](#) Sets the resistance of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoHalfEdge Class

[Classes](#)

Manages topology half edges. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoHalfEdge;
```

File

AcMapTopoHalfEdge.h

☐ Methods



[~AcMapTopoHalfEdge](#) Destroys an instance of this class.

[GetFullEdge](#) Retrieves the full edge that contains this half edge.

[GetNextEdge](#) Retrieves the next edge by turning left or right at the next node.

[GetNextNode](#) Retrieves the next node by moving forward along this edge.

[GetPolygon](#) Retrieves the polygon associated with this edge (default level).

[GetPreviousNode](#) Retrieves the previous node by moving backward along this edge.

[GetResistance](#) Retrieves the resistance of this edge.

[GetRing](#) Retrieves the ring associated with this edge (default level).

[GetTopology](#) Retrieves the topology that contains this edge.

[SetResistance](#) Sets the resistance of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoIterator Class

[Classes](#)

An iterator over a collection of AcMapTopologyinstances. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoIterator;
```

File

AcMapTopoIterator.h

☐ Methods



[~AcMapTopoIterator](#) Destroys an instance of this class.



[AcMapTopoIterator](#) Constructs an instance of this class.

[Count](#) Counts the number of elements in the iteration.

[First](#) Moves to the first element in the iteration.

[GetDescription](#) Retrieves the description of this topology model.

[GetName](#) Retrieves the name of this topology model.

[GetTopology](#) Retrieves the topology of the current element in the iteration.

[GetType](#) Retrieves the type of this topology model.

[IsDone](#) Determines whether the iterator has reached the end of the collection.

[Next](#) Advances to the next element in the iteration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopology Class

[Classes](#)

The main topology object. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopology;
```

File

AcMapTopology.h

☐ Enumerations

- ☞ [EAuditResults](#) Enumerates the result types of an audit.
- ☞ [ECreateOptions](#) Enumerates the options for creating a topology.
- ☞ [EOpenMode](#) Enumerates the ways to open a topology model.
- ☞ [EStatus](#) Enumerates the status types of a topology model.

☐ Methods

- ☞ [~AcMapTopology](#) Destroys an instance of this class.
- ☞ [AcMapTopology](#) Constructs an instance of this class.
- [AddCurveObject](#) Adds a curve (linear object) to this topology model.
- [AddPointObject](#) Adds a point to this topology model.
- [AddPolygons](#) Adds objects to a polygon topology model.
- [Buffer](#) Creates a new topology with the specified buffer setting on the current topology model. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).
- [Clip](#) Uses the overlay polygon topology as a boundary to clip and discard the parts of the source polygons outside the overlay polygons. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#),

	<a href="#">SetEdgeCreationSettings()</a> , and <a href="#">SetCentroidCreationSettings()</a> .
<a href="#">Close</a>	Closes this topology model. Creates a topology. To set topology-creation values, call the three functions <a href="#">SetNodeCreationSettings()</a> , <a href="#">SetEdgeCreationSettings()</a> , and <a href="#">SetCentroidCreationSettings()</a> . If caller does not call these functions, then node creation defaults to false, missing-centroid creation defaults to true, and the node or centroid block name defaults to ACAD_POINT.
<a href="#">Create</a>	Creates a topology - this function is not yet implemented in AutoCAD Map; use the other <a href="#">Create()</a> function.
<a href="#">DeleteEdge</a>	Deletes an edge from this topology model.
<a href="#">DeleteNode</a>	Deletes a node from this topology model.
<a href="#">DeleteNode</a>	Deletes a node from this topology model.
<a href="#">DeletePolygon</a>	Deletes a polygon from the current topology model.
<a href="#">DeletePolygon</a>	Deletes a polygon from the current topology model.
<a href="#">Dissolve</a>	Combines entities that share the same data value in a specified column. Uses the overlay polygon topology as a mask to erase everything in the source polygon topology that is covered by the overlay topology. To set topology-creation values, call the three functions <a href="#">SetNodeCreationSettings()</a> , <a href="#">SetEdgeCreationSettings()</a> , and <a href="#">SetCentroidCreationSettings()</a> .
<a href="#">Erase</a>	Uses the overlay polygon topology as a mask to erase everything in the source polygon topology that is covered by the overlay topology. To set topology-creation values, call the three functions <a href="#">SetNodeCreationSettings()</a> , <a href="#">SetEdgeCreationSettings()</a> , and <a href="#">SetCentroidCreationSettings()</a> .
<a href="#">FindEdge</a>	Finds the edge closest to a specified point.
<a href="#">FindEdge</a>	Finds the edge with the specified ID.
<a href="#">FindNeighborPolygons</a>	Finds all polygons that are neighbors of those edges that are based on a specified curve.
<a href="#">FindNode</a>	Finds the node nearest a specified point.

<a href="#"><u>FindPolygon</u></a>	Finds the smallest polygon containing a specified point.
<a href="#"><u>FindTopologyObject</u></a>	Finds the object (node, full edge, or polygon) nearest a specified point within a specified tolerance.
<a href="#"><u>GetBackwardEdge</u></a>	Retrieves the half edge with the specified object ID from this topology model.
<a href="#"><u>GetDescription</u></a>	Retrieves the description of this topology model.
<a href="#"><u>GetEntityId</u></a>	Retrieves the object ID of the AutoCAD entity associated with the specified topology element ID in the current drawing or source drawing.
<a href="#"><u>GetEntityIds</u></a>	Retrieves the object IDs of all the AutoCAD entities associated with the topology elements in the current drawing.
<a href="#"><u>GetForwardEdge</u></a>	Retrieves the half edge with the specified object ID from this topology model.
<a href="#"><u>GetFullEdge</u></a>	Retrieves the full edge with the specified object ID from this topology model.
<a href="#"><u>GetFullEdges</u></a>	Retrieves all full edges from this topology model.
<a href="#"><u>GetMarkerStyles</u></a>	Retrieves the pointer to the error-marker settings of this topology model.
<a href="#"><u>GetName</u></a>	Retrieves the name of this topology model.
<a href="#"><u>GetNode</u></a>	Retrieves the node with the specified object ID from this topology model.
<a href="#"><u>GetNodes</u></a>	Retrieves all nodes from this topology model.
<a href="#"><u>GetPolygon</u></a>	Retrieves the polygon with the specified object ID from this topology model.
<a href="#"><u>GetPolygons</u></a>	Retrieves all polygons from this topology model.
<a href="#"><u>GetStatus</u></a>	Retrieves the status of this topology model.
<a href="#"><u>GetType</u></a>	Retrieves the type of this topology model.
<a href="#"><u>Identity</u></a>	Performs a Union() on the source topology and an Intersect() on the overlay topology. To set topology-creation values, call the three functions <a href="#"><u>SetNodeCreationSettings()</u></a> , <a href="#"><u>SetEdgeCreationSettings()</u></a> , and <a href="#"><u>SetCentroidCreationSettings()</u></a> .

<a href="#"><u>Intersect</u></a>	Combines topologies and keeps only the common geometry. Intersect acts like the Boolean AND operation. The results are the same no matter the order topologies are chosen. Object data are combined for the two operations. To set topology-creation values, call the three functions <a href="#"><u>SetNodeCreationSettings()</u></a> , <a href="#"><u>SetEdgeCreationSettings()</u></a> , and <a href="#"><u>SetCentroidCreationSettings()</u></a> .
<a href="#"><u>IsComplete</u></a>	Determines whether this topology model is complete.
<a href="#"><u>IsFixedType</u></a>	Determines whether this topology model can contain only elements of its own type.
<a href="#"><u>IsLinearType</u></a>	Determines whether this topology model is a network topology.
<a href="#"><u>IsLogicalType</u></a>	Determines whether the elements in this topology model can be connected if their graphical representations are farther apart than the tolerance value.
<a href="#"><u>IsPointType</u></a>	Determines whether this topology model is a point topology.
<a href="#"><u>IsPolygonType</u></a>	Determines whether this topology model is a polygon topology.
<a href="#"><u>MergePolygons</u></a>	Merges neighboring polygons, dissolving the boundaries between them.
<a href="#"><u>MergePolygons</u></a>	Merges neighboring polygons, dissolving the boundaries between them.
<a href="#"><u>MergePolygons</u></a>	Merges two neighboring polygons, dissolving the boundaries between them.
<a href="#"><u>MergePolygons</u></a>	Merges two neighboring polygons, dissolving the boundaries between them.
<a href="#"><u>MoveNode</u></a>	Moves a node to the new position.
<a href="#"><u>MoveNode</u></a>	Moves a node to the new position.
<a href="#"><u>NeedsRefresh</u></a>	Determines whether this topology model is correct.

<a href="#">Open</a>	Opens the topology model in only the current drawing. After you are finished with a topology model, call <code>Close()</code> to close it.
<a href="#">Open</a>	Opens the topology model in both current and source (attached) drawings. After you are finished with a topology model, call <code>Close()</code> to close it.
<a href="#">Paste</a>	Pastes the overlay polygon topology on top of the source polygons. The source polygons not covered by the overlay remain. Paste can be used with only polygons. To set topology-creation values, call the three functions <a href="#">SetNodeCreationSettings()</a> , <a href="#">SetEdgeCreationSettings()</a> , and <a href="#">SetCentroidCreationSettings()</a> .
<a href="#">Refresh</a>	Updates a topology element.
<a href="#">SetCentroidCreationSettings</a>	Sets the centroid-creation parameters of this topology model.
<a href="#">SetDescription</a>	Sets the description of this topology model.
<a href="#">SetEdgeCreationSettings</a>	Sets the edge-creation parameters of this topology model.
<a href="#">SetNodeCreationSettings</a>	Sets the node-creation parameters of this topology model.
<a href="#">ShowGeometry</a>	Highlights the topology model with the specified color.
<a href="#">SplitPolygon</a>	Splits a polygon along a curve.
<a href="#">SplitPolygon</a>	Splits a polygon along a curve.
<a href="#">TraceBestPath</a>	Finds the least-cost path between start and end points with intermediate points to visit.
<a href="#">TraceFlood</a>	Traces the flood from a specified point.
<a href="#">TraceLeastCostPath</a>	Finds the least-cost path between two points.
<a href="#">Union</a>	Combines polygons with polygons and keeps all geometry. Union acts like the Boolean OR operation and can be used with only polygons. To set topology-creation values, call the three functions <a href="#">SetNodeCreationSettings()</a> , <a href="#">SetEdgeCreationSettings()</a> , and

[SetCentroidCreationSettings\(\)](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)  
AcMapTopology:: AcMapTopology Constructor  
[AcMapTopology Class](#) | [AcMapTopology Class](#)

Constructs an instance of this class.

```
AcMapTopology(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input name of the new topology.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: Buffer Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Creates a new topology with the specified buffer setting on the current topology model. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrCode Buffer(  
    const ACHAR* pszOffset,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    double defaultDistance = 1.0  
);
```

Parameters	Description
pszOffset	Input offset distance, as a string. Set this value to NULL to use the default value.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
defaultDistance	Input default distance of the new topology. The default value is 1.0. The measurement units depend on the current drawing's unit settings.

## Returns

Returns `AcMap::EErrCode kOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::Clip Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Uses the overlay polygon topology as a boundary to clip and discard the parts of the source polygons outside the overlay polygons. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrorCode Clip(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: Dissolve Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Combines entities that share the same data value in a specified column.

```
AcMap::EErrCode Dissolve(  
    const ACHAR* pszAttributeExpr,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataField* pResultDataField = NULL  
);
```

Parameters	Description
pszAttributeExpr	Input attribute expression string.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
pResultDataField	Input <a href="#">AcMapObjectDataField</a> result object data table column settings. The default value is NULL.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

## Remarks

When you dissolve a topology, AutoCAD Map checks each boundary between entities to see if the dissolve column value is the same for both. If it is, the boundary and one of the entities are removed. All object data columns except the dissolve and any new topology columns are removed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::Erase Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Uses the overlay polygon topology as a mask to erase everything in the source polygon topology that is covered by the overlay topology. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrCode Erase(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::Identity Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Performs a Union() on the source topology and an Intersect() on the overlay topology. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrCode Identity(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::Intersect Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Combines topologies and keeps only the common geometry. Intersect acts like the Boolean AND operation. The results are the same no matter the order topologies are chosen. Object data are combined for the two operations. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrorCode Intersect(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology . The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns `AcMap::EErrorCode kOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: Paste Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Pastes the overlay polygon topology on top of the source polygons. The source polygons not covered by the overlay remain. Paste can be used with only polygons. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrorCode Paste(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description of the new topology. The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: setDescription Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Sets the description of this topology model.

```
AcMap::EErrorCode setDescription(  
    const ACHAR* pszName  
);
```

Parameters	Description
pszName	Input description of the topology model.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: Union Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Combines polygons with polygons and keeps all geometry. Union acts like the Boolean OR operation and can be used with only polygons. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#).

```
AcMap::EErrCode Union(  
    const AcMapTopology*& pOverlayTopology,  
    const ACHAR* pszTopoName,  
    const ACHAR* pszTopoDesc = NULL,  
    const AcMapObjectDataTable* pResultDataTable = NULL,  
    const AcMapOverlayDataArray* paSourceData = NULL,  
    const AcMapOverlayDataArray* paOverlayData = NULL  
);
```

Parameters	Description
pOverlayTopology	Input overlay topology pointer.
pszTopoName	Input name of the new topology.
pszTopoDesc	Input description the new topology. The default value is NULL.
pResultDataTable	Input <a href="#">AcMapObjectDataTable</a> result object data table settings. The default value is NULL.
paSourceData	Input source data settings. The default value is NULL.
paOverlayData	Input overlay data settings. The default value is NULL.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopologySource Class

[Classes](#)

Stores the topology source information. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopologySource;
```

File

AcMapTopologyManager.h

☐ Methods



[~AcMapTopologySource](#) Destroys an instance of this class.



[AcMapTopologySource](#) Constructs an instance of this class.



[AcMapTopologySource](#) Constructs an instance of this class by using a copy constructor.

[GetName](#) Retrieves the topology name.

[GetPath](#) Retrieves the full path of the topology.

[GetSource](#) Retrieves the source of the topology.

[IsLoaded](#) Determines whether the topology is loaded.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoNode Class

[Classes](#)

Manages topology nodes. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoNode : public AcMapTopoElement;
```

File

AcMapTopoNode.h

☐ Methods

<a href="#">~AcMapTopoNode</a>	Destroys an instance of this class.
<a href="#">GetEdges</a>	Retrieves this node's edges in counter-clockwise order.
<a href="#">GetEntity</a>	Retrieves the object identifier of the AutoCAD entity associated with this node.
<a href="#">GetID</a>	Retrieves the unique identifier of this node in its topology.
<a href="#">GetLocation</a>	Retrieves the position of this node.
<a href="#">GetNextEdge</a>	Retrieves the next edge given an incoming full edge and a left or right turn direction.
<a href="#">GetNextEdge</a>	Retrieves the next edge given an incoming half edge and a left or right turn direction.
<a href="#">GetResistance</a>	Retrieves the resistance of this node.
<a href="#">GetTopology</a>	Retrieves the topology that contains this node.
<a href="#">IsMoveable</a>	Determines whether this node is moveable - this function is not yet implemented in AutoCAD Map and always returns true.
<a href="#">IsOnThisObject</a>	Determines whether a specified point is within tolerance of this node.
<a href="#">SetIsMoveable</a>	Sets whether this node is moveable - this function is not yet implemented in AutoCAD Map.
<a href="#">SetLocation</a>	Sets the position of this node - this function is not yet implemented in AutoCAD Map.

## [SetResistance](#)

Sets the resistance of this node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoOverlayData Class

[Classes](#)

For topology overlay analysis, use this class to set the data attributes that contain object data from linked database to include in the resulting topology, and the name and description of the new object data table. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoOverlayData;
```

File

AcMapTopology.h

☐ Methods



[~AcMapTopoOverlayData](#) Destroys an instance of this class.



[AcMapTopoOverlayData](#) Constructs an instance of this class.



[AcMapTopoOverlayData](#) Constructs an instance of this class by using the specified expression, column name, and column data type.



[AcMapTopoOverlayData](#) Constructs an instance of this class by using a copy constructor.

[GetColumnName](#) Retrieves the column name of the object data table.

[GetColumnType](#) Retrieves the column data type of the object data table.

[GetExpression](#) Retrieves the expression.

[SetColumnName](#) Sets the column name of the object data table.

[SetColumnType](#) Sets the column data type of the object data table.

[SetExpression](#) Sets the expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)

[AcMapTopoOverlayData::AcMapTopoOverlayData Constructor](#)

[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Constructs an instance of this class by using the specified expression, column name, and column data type.

```
AcMapTopoOverlayData(  
    const ACHAR* pszExpression,  
    const ACHAR* pszColumnName,  
    AcMap::EDataType enmColumnType  
);
```

Parameters	Description
pszExpression	Input the expression.
pszColumnName	Input column name.
enmColumnType	Input column data type.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
AcMapTopoOverlayData:: SetColumnName Method  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Sets the column name of the object data table.

```
AcMap::EErrCode SetColumnName(  
    const ACHAR* pszColumnName  
);
```

Parameters	Description
pszColumnName	Input column name.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
[AcMapTopoOverlayData:: SetExpression Method](#)  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Sets the expression.

```
AcMap::EErrCode SetExpression(  
    const ACHAR* pszExpression  
);
```

Parameters	Description
pszExpression	Input expression.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoPolygon Class

[Classes](#)

Manages topology polygons. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoPolygon : public AcMapTopoElement;
```

File

AcMapTopoPolygon.h

☐ Methods



<a href="#">~AcMapTopoPolygon</a>	Destroys an instance of this class.
<a href="#">GetArea</a>	Retrieves the area of this polygon.
<a href="#">GetBoundary</a>	Retrieves the boundary of this polygon, represented by a list of rings.
<a href="#">GetCentroid</a>	Retrieves the position of this polygon's centroid.
<a href="#">GetEntity</a>	Retrieves the object identifier of the AutoCAD entity associated with this polygon.
<a href="#">GetHierChildren</a>	Retrieves the hierarchical child polygons, if any, of this polygon - this function is not yet implemented in AutoCAD Map.
<a href="#">GetHierParent</a>	Retrieves the hierarchical parent polygon, if any, of this polygon - this function is not yet implemented in AutoCAD Map.
<a href="#">GetID</a>	Retrieves the unique identifier of this polygon in its topology.
<a href="#">GetParent</a>	Retrieves the multi-polygon parent, if any, of this polygon - this function is not yet implemented in AutoCAD Map.
<a href="#">GetPerimeter</a>	Retrieves the perimeter of this polygon.
<a href="#">GetTopology</a>	Retrieves the topology that contains this polygon.
<a href="#">IsOnThisObject</a>	Determines whether a specified point is within tolerance of this polygon.

## Traverse

Traverses to another polygon by following a curve -  
this function is not yet implemented in AutoCAD Map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTopoRing Class

[Classes](#)

Manages topology rings. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTopoRing;
```

File

AcMapTopoRing.h

☐ Methods

<a href="#">~AcMapTopoRing</a>	Destroys an instance of this class.
<a href="#">GetArea</a>	Retrieves the area of this ring.
<a href="#">GetEdges</a>	Retrieves the edges that make up this ring.
<a href="#">GetLength</a>	Retrieves the length of this ring.
<a href="#">GetStartEdge</a>	Retrieves the first edge in this ring.
<a href="#">IsExterior</a>	Determines whether this ring is exterior (that is, an outer boundary of its polygon).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Classes](#)

AcMapTraceParameters Class

[Classes](#)

Settings for trace parameters. For more information, search for *topology* in AutoCAD Map Help.

```
class AcMapTraceParameters : public AcMapNetAnalysisParameters;  
File
```

AcMapTopology.h

☐ Methods



[~AcMapTraceParameters](#) Destroys an instance of this class.



[AcMapTraceParameters](#) Constructs an instance of this class.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTraceParameters Class](#), [AcMapTraceParameters Class](#)

[AcMapTraceParameters:: AcMapTraceParameters Constructor](#)

[AcMapTraceParameters Class](#) | [AcMapTraceParameters Class](#)

Constructs an instance of this class.

```
AcMapTraceParameters(  
    double dMinResistance,  
    double dMaxResistance,  
    const ACHAR* aNodeResistanceExpression,  
    const ACHAR* aLinkDirectionExpression,  
    const ACHAR* aLinkResistanceExpression,  
    const ACHAR* aLinkReverseResistanceExpression,  
    bool bUseReverseDirection,  
    const ACHAR* aName = NULL,  
    const ACHAR* aDescription = NULL  
);
```

Parameters	Description
dMinResistance	Input minimum resistance.
dMaxResistance	Input maximum resistance.
aNodeResistanceExpression	Input node resistance expression.
aLinkDirectionExpression	Input link direction expression.
aLinkResistanceExpression	Input link resistance expression.
aLinkReverseResistanceExpression	Input link reverse resistance expression.
bUseReverseDirection	Input true to use reverse direction; otherwise, false.
aName	Input name of the new topology. If this value is NULL, no new topology is created. The default value is NULL.
aDescription	Input description of the new topology. The default value is NULL.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[Topology: Class-Based API](#), [Structures](#)

Structs, Records, Enums

[Topology: Class-Based API](#) | [Structures](#)

☐ Enumerations



[ETopologyMarkType](#) Topology error-marker shapes.



[ETopologyType](#) Topology types.

☐ Structures



[AcMapObjectDataField](#) Information about a field in a topology object data table.



[AcMapObjectDataTable](#) Information about a topology object data table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Structs, Records, Enums](#)

AcMapObjectDataField Structure

[Structs, Records, Enums](#)

Information about a field in a topology object data table.

```
struct AcMapObjectDataField {  
    ACHAR m_aODFieldName[MAX_FIELD_NAME];  
    ACHAR m_aODTableName[MAX_TABLE_NAME];  
};
```

File

AcMapTopologyCommonDef.h

Parameters	Description
m_aODFieldName[MAX_FIELD_NAME]	Field name. Default MAX_FIELD_NAME is 32.
m_aODTableName[MAX_TABLE_NAME]	Table name. Default MAX_TABLE_NAME is 25.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Structs, Records, Enums](#)

AcMapObjectDataTable Structure

[Structs, Records, Enums](#)

Information about a topology object data table.

```
struct AcMapObjectDataTable {  
    ACHAR m_aODTableName[MAX_TABLE_NAME];  
    ACHAR m_aODTableDescription[MAX_DESC];  
};
```

File

AcMapTopologyCommonDef.h

Parameters	Description
m_aODTableName[MAX_TABLE_NAME]	Table name. Default MAX_TABLE_NAME is 25.
m_aODTableDescription[MAX_DESC]	Table description. Default MAX_DESC is 512.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Structs, Records, Enums](#)

ETopologyMarkType Enumeration

[Structs, Records, Enums](#)

Topology error-marker shapes.

```
enum ETopologyMarkType {  
    eMarkOctagon = 1,  
    eMarkTriangle = 2,  
    eMarkRhombus = 3,  
    eMarkSquare = 4  
};
```

File

AcMapTopologyCommonDef.h

Parameters	Description
eMarkOctagon	Octagon.
eMarkTriangle	Triangle.
eMarkRhombus	Rhombus.
eMarkSquare	Square.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Structs, Records, Enums](#)

ETopologyType Enumeration

[Structs, Records, Enums](#)

Topology types.

```
enum ETopologyType {  
    ePoint = 0x01,  
    eLinear = 0x02,  
    ePolygon = 0x04,  
    eLogical = 0x08,  
    eFixed = 0x10  
};
```

File

AcMapTopologyCommonDef.h

Parameters	Description
ePoint	Node topology.
eLinear	Network topology.
ePolygon	Polygon topology.
eLogical	Geometry/topology mismatch (electrical utility style). Always false.
eFixed	Topology cannot accept elements of other types. Always true.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Topology: Class-Based API](#)

Types

[Topology: Class-Based API](#)

▣ Types

[AcMapFullEdgePtrArray](#)

Dynamic array of AcMapTopoFullEdgepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapHalfEdgePtrArray](#)

Dynamic array of AcMapTopoHalfEdgepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapNodePtrArray](#)

Dynamic array of AcMapTopoNodepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapObjectPtrArray](#)

Dynamic array of AcMapTopoElementpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapOverlayDataArray](#)

This is type AcMapOverlayDataArray.

[AcMapPolygonPtrArray](#)

Dynamic array of AcMapTopoPolygonpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapRingPtrArray](#)

Dynamic array of AcMapTopoRingpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

[AcMapTopologySourceArray](#) This is type AcMapTopologySourceArray.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapFullEdgePtrArray Type

[Types](#)

Dynamic array of AcMapTopoFullEdgepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoFullEdge> AcMapFullEdgePtr  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapHalfEdgePtrArray Type

[Types](#)

Dynamic array of AcMapTopoHalfEdgepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoHalfEdge> AcMapHalfEdgePtr  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapNodePtrArray Type

[Types](#)

Dynamic array of AcMapTopoNodepointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoNode> AcMapNodePtrArray;  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapObjectPtrArray Type

[Types](#)

Dynamic array of AcMapTopoElementpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoElement> AcMapObjectPtrArr  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapOverlayDataArray Type

[Types](#)

This is type AcMapOverlayDataArray.

```
typedef AcArray< AcMapTopoOverlayData > AcMapOverlayDataArray;  
File
```

AcMapTopology.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapPolygonPtrArray Type

[Types](#)

Dynamic array of AcMapTopoPolygonpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoPolygon> AcMapPolygonPtrAr  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapRingPtrArray Type

[Types](#)

Dynamic array of AcMapTopoRingpointers. For information about member functions, see the template class [AcMapTopoElementPtrArray](#) and its parent class AcArray.

```
typedef AcMapTopoElementPtrArray<AcMapTopoRing> AcMapRingPtrArray;  
File
```

AcMapTopologyCommonDef.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[Types](#)

AcMapTopologySourceArray Type

[Types](#)

This is type AcMapTopologySourceArray.

```
typedef AcArray< AcMapTopologySource > AcMapTopologySourceArray;  
File
```

AcMapTopologyManager.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Topology Global-Function API (Deprecated)

Except for [drawing cleanup](#) functions, the global-function API for topology is deprecated in favor of the [class-based API](#).

<a href="#">Topology Function Catalog</a>	Topology functions sorted by name.
<a href="#">Topology Function Synopsis</a>	Topology functions sorted by functional group.
<a href="#">Topology Configuration Variables</a>	Cleanup variables, cleanup-action variables, topology variables.



# Topology Function Catalog

[Topology Global-Function API](#)

---

Topology functions sorted by name.

**Note** Except for [drawing cleanup](#) functions, the global-function API for topology is deprecated in favor of the [class-based API](#).

[ac](#) | [ana](#) | [clean](#) | [edit](#) | [elem](#) | [info](#) | [iter](#) | [mnt](#) | [qry](#) | [trace](#) | [var](#)

[tpm\\_acclose](#) [tpm\\_acexist](#)

[tpm\\_acload](#)

[tpm\\_acopen](#)

[tpm\\_acqty](#)

[tpm\\_acunload](#)

[tpm\\_acupgradeopen](#)

[tpm\\_anabuffer](#)

[tpm\\_anadissolve](#)

[tpm\\_anaoverlay](#)

[tpm\\_cleanactionlistdel](#)

[tpm\\_cleanactionlistgetat](#)

[tpm\\_cleanactionlistins](#)

[tpm\\_cleanactionlistqty](#)

[tpm\\_cleanalloc](#)

[tpm\\_cleananchorss](#)

[tpm\\_cleancancel](#)

[tpm\\_cleancomplete](#)

[tpm\\_cleancreatedss](#)

[tpm\\_cleanend](#)

[tpm\\_cleanerrorcur](#)

[tpm\\_cleanerrordraw](#)

[tpm\\_cleanerrorfix](#)

[tpm\\_cleanerrorget](#)

[tpm\\_cleanerrormark](#)

[tpm\\_cleanerrorset](#)

[tpm\\_cleanfree](#)

[tpm\\_cleangroupdraw](#)

[tpm\\_cleangroupfix](#)  
[tpm\\_cleangroupmark](#)  
[tpm\\_cleangroupnext](#)  
[tpm\\_cleangroupqty](#)  
[tpm\\_cleangroupsubtype](#)  
[tpm\\_cleangrouptype](#)  
[tpm\\_cleanincludess](#)  
[tpm\\_cleaninit](#)  
[tpm\\_cleaninitanchorset](#)  
[tpm\\_cleanmodifiedss](#)  
[tpm\\_cleanprofileload](#)  
[tpm\\_cleanprofilesave](#)  
[tpm\\_cleanstart](#)  
[tpm\\_cleanunchangedss](#)

[tpm\\_editaddelem](#)  
[tpm\\_editdelem](#)  
[tpm\\_editmodelem](#)  
[tpm\\_editupdelem](#)

[tpm\\_elemadj](#)  
[tpm\\_elemfind](#)  
[tpm\\_elemget](#)  
[tpm\\_elemid](#)  
[tpm\\_elemqty](#)  
[tpm\\_elemss](#)

[tpm\\_infobuildvar](#)  
[tpm\\_infocomplete](#)  
[tpm\\_infocorrect](#)  
[tpm\\_infocurrent](#)  
[tpm\\_infodesc](#)  
[tpm\\_infomodified](#)  
[tpm\\_infoname](#)  
[tpm\\_infostatus](#)  
[tpm\\_infotype](#)  
[tpm\\_infoversion](#)

[tpm\\_iterdesc](#)  
[tpm\\_itername](#)  
[tpm\\_iternext](#)  
[tpm\\_iterstart](#)  
[tpm\\_iterstop](#)  
[tpm\\_itype](#)  
[tpm\\_itypeversion](#)

[tpm\\_mntbuild](#)  
[tpm\\_mnterase](#)  
[tpm\\_mntrebuild](#)  
[tpm\\_mntrename](#)

[tpm\\_qrygetresdesc](#)  
[tpm\\_qrygetrestopo](#)  
[tpm\\_qrygettoponame](#)  
[tpm\\_qrysetrestopo](#)  
[tpm\\_qrysettoponame](#)

[tpm\\_tracealloc](#)  
[tpm\\_tracebestroute](#)  
[tpm\\_tracebestroutescan](#)  
[tpm\\_tracebestrouteval](#)  
[tpm\\_traceelemedit](#)  
[tpm\\_traceelemget](#)  
[tpm\\_traceelemid](#)  
[tpm\\_traceflood](#)  
[tpm\\_tracefree](#)  
[tpm\\_traceqty](#)  
[tpm\\_tracesetmaxres](#)  
[tpm\\_tracesetminres](#)  
[tpm\\_traceshort](#)  
[tpm\\_traceshortscan](#)  
[tpm\\_traceshortval](#)

[tpm\\_varalloc](#)  
[tpm\\_varfree](#)  
[tpm\\_varget](#)  
[tpm\\_varlist](#)  
[tpm\\_varset](#)



# Topology Function Synopsis

[Topology Global-Function API](#)

---

Topology functions sorted by functional group.

**Note** Except for [drawing cleanup](#) functions, the global-function API for topology is deprecated in favor of the [class-based API](#).

[Access Functions](#) [Analyzing Functions](#)

[Building and Erasing Functions](#)

[Drawing Cleanup Functions](#)

[Editing Functions](#)

[Element Information Functions](#)

[Iterating Functions](#)

[Network Tracing Functions](#)

[Topology Information Functions](#)

[Topology Query Functions](#)

[Topology Variables Functions](#)



## Cleanup Variables

### [Topology Configuration Variables](#)

---

Cleanup variables store properties for cleanup models.

Cleanup variables are a subset of the [configuration variables](#) data structure.

**Note** Data types cited below are AutoLISP types. The AutoLISP **real** corresponds to the C++ **double**. Similarly, **integer** corresponds to **long**.

<b>ANCHOROBJS_FEATURES</b>	<b>!! New !!</b> Classified objects to include, a comma-separated list of feature names ( <b>string</b> ), such as "Roads, Rivers, Streets". Default = "*" (all features).
<a href="#">ANCHOROBJS_LAYERS</a>	Anchored entities specified by <a href="#">tpm_cleaninitanchorset</a> will be anchored only if they reside on layers specified here, with multiple layer names separated by commas. For example, "0,Layer1,Layer2". Default = "*" (any layer)
<b>APPARENT_INTERSECTION_COLOR</b>	<b>!! New !!</b> Color for apparent-intersection markers, an AutoCAD color index ( <b>integer</b> ). See <a href="#">Color Index Colors</a> . Default = 7 (white)
<b>APPARENT_INTERSECTION_MARKER</b>	<b>!! New !!</b> Marker shape for apparent-intersection errors ( <b>integer</b> ). See <a href="#">Marker Shapes</a> . Default = 2 (triangle)
<b>ARC_TYPE</b>	Convert arcs to ( <b>integer</b> ): 0 = Arc (default) 1 = 2D polyline
<b>CIRCLE_TYPE</b>	Convert circles to ( <b>integer</b> ): 0 = Circle (default) 1 = 2D polyline 2 = Arc
<b>CLEAN_TOL</b>	Cleanup tolerance (positive <b>real</b> ). Default = 0.01
<b>CLUSTER_COLOR</b>	Color for cluster markers, an AutoCAD color index ( <b>integer</b> ). See <a href="#">Color Index Colors</a> .

	Default = 5 (blue)
CLUSTER_MARKER	Marker shape for cluster errors ( <i>integer</i> ). See <a href="#">Marker Shapes</a> . Default = 4 (square)
CONVERT	Flag for the original entities conversion ( <i>integer</i> ). 0 = Do not convert (default) 1 = Convert
CORRIDOR	Corridor width (positive <i>real</i> ). Default = 0.01 <b>Note</b> If you are using an explicit cleanup action list (specifying cleanup actions using <a href="#">tpm_cleanactionlistins</a> ), this setting is ignored.
CROSS_COLOR	Color for cross markers, an AutoCAD color index ( <i>integer</i> ). See <a href="#">Color Index Colors</a> . Default = 2 (yellow)
CROSS_MARKER	Marker shape for cross errors ( <i>integer</i> ). See <a href="#">Marker Shapes</a> . Default = 1 (octagon)
DANGL_COLOR	Color for dangling node markers, an AutoCAD color index ( <i>integer</i> ). See <a href="#">Color Index Colors</a> . Default = 1 (red)
DANGL_MARKER	Marker shape for dangling node errors ( <i>integer</i> ). See <a href="#">Marker Shapes</a> . Default = 1 (octagon)
DEL_MARKER	Whether to delete previous error markers ( <i>integer</i> ). 0 = Do not delete 1 = Delete (default)
DUPL_COLOR	Color for duplicates markers, an AutoCAD color index ( <i>integer</i> ). See <a href="#">Color Index Colors</a> . Default = 6 (magenta)
DUPL_MARKER	Marker shape for duplicate errors ( <i>integer</i> ). See <a href="#">Marker Shapes</a> . Default = 1 (octagon)
ENT_PROCESS	Flag for original entities processing ( <i>integer</i> ). 1 = Convert original 2 = Create new and keep original 3 = Create new and delete original Default = 0

GENERALIZE	<p>Flag for generalization (link simplification) (<i>integer</i>).</p> <p>0 = Do not generalize (default)</p> <p>1 = Generalize</p> <p><b>Note</b> If you are using an explicit cleanup action list (specifying cleanup actions using <a href="#">tpm_cleanactionlists</a>), this setting is ignored.</p>
INCLUDE_LINEAROBJS	<p><b>!! New !!</b> Whether to include linear objects while deleting duplicates (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 1.</p>
INCLUDE_POINTS	<p><b>!! New !!</b> Whether to include points while deleting duplicates or snapping clustered nodes (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 1.</p>
INCLUDE_BLOCKS	<p><b>!! New !!</b> Whether to include blocks while deleting duplicates or snapping clustered nodes (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 0.</p>
INCLUDE_TEXT	<p><b>!! New !!</b> Whether to include text while deleting duplicates (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 0.</p>
INCLUDE_MTEXT	<p><b>!! New !!</b> Whether to include mtext while deleting duplicates (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 1.</p>
INCLUDE_ROTATION	<p><b>!! New !!</b> Whether to include rotation while deleting duplicates (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p> <p>Default = 0.</p>
INCLUDE_ZVALUES	<p><b>!! New !!</b> Whether to include z-values while deleting duplicates (<i>integer</i>).</p> <p>0 = Do not include.</p> <p>1 = Include.</p>

	Default = 0.
SNAP_TO_NODE	<p><b>!! New !!</b> Whether to snap to nodes or links (<b>integer</b>).</p> <p>0 = Snap to nodes. 1 = Snap to links. Default = 0.</p>
<a href="#">INCLUDEOBS AUTOSELECT</a>	<p>How entities are specified for cleanup</p> <p>0 = Select entities manually (that is, by passing a selection set to <a href="#">tpm_cleaninit</a>)</p> <p>1 = Select all entities in the drawing (in which case the selection set passed to <a href="#">tpm_cleaninit</a>, if any, is ignored)</p>
INCLUDEOBS_FEATURES	<p><b>!! New !!</b> Classified objects to include, a comma-separated list of features (<b>string</b>), such as "Roads, Rivers, Streets". Default = "*" (all features).</p>
<a href="#">INCLUDEOBS LAYERS</a>	<p>Entities specified by <a href="#">tpm_cleaninit</a> will be cleaned only if they reside on layers specified here, with multiple layer names separated by commas. For example, "0,Layer1,Layer2". Default = "*" (any layer)</p>
LINE_TYPE	<p>Convert lines to (<b>integer</b>)</p> <p>0 = Line (default) 1 = 2D polyline</p>
LINK_COLOR	<p>Color for new links, an AutoCAD color index (<b>integer</b>). See <a href="#">Color Index Colors</a>.</p> <p>Negative <b>integer</b> = Current color Default = -1</p>
LINK_CORRECT	<p>Flag for link error correction (<b>integer</b>).</p> <p>0 = Do not correct 1 = Correct (default)</p> <p><b>Note</b> If you are using an explicit cleanup action list (specifying cleanup actions using <a href="#">tpm_cleanactionlistins</a>), this setting is ignored.</p>
LINK_ERROR	<p>Link error types (<b>integer</b>), sum of the desired options.</p> <p>0 = None 1 = Short 2 = Cross 4 = Undershoot 8 = Duplicate Default = 15</p>
LINK_LAYER	<p>Layer name for new links (<b>string</b>).</p> <p>nil = Current layer</p>

	Default = ""
<a href="#">MAINTAIN_MARKERS</a>	<p>Flag to maintain markers when cleanup ends</p> <p>0 = Do not maintain</p> <p>1 = Maintain markers</p> <p>This variable affects only errors that have not been fixed. If an error has been fixed, its marker is deleted at the end of the cleanup process no matter what (by <a href="#">tpm_cleanend</a>).</p>
<a href="#">MARKER_HEIGHT</a>	<p>The height of error markers, a percent of screen height (positive <b>real</b>).</p> <p>Default = 5.0</p>
<a href="#">NODE_CORRECT</a>	<p>Flag for node error correction (<b>integer</b>).</p> <p>0 = Do not correct</p> <p>1 = Correct (default)</p> <p><b>Note</b> If you are using an explicit cleanup action list (specifying cleanup actions using <a href="#">tpm_cleanactionlistins</a>), this setting is ignored.</p>
<a href="#">NODE_ERROR</a>	<p>Node error types (<b>integer</b>), sum of the desired options.</p> <p>0 = None</p> <p>16 = Cluster</p> <p>32 = Pseudo</p> <p>64 = Dangling</p> <p>Default = 48</p>
<a href="#">POLY3D_TYPE</a>	<p>Convert 3D polylines to (<b>integer</b>).</p> <p>0 = 3D polyline (default)</p> <p>1 = 2D polyline</p>
<a href="#">PSEUDO_COLOR</a>	<p>Color for pseudo node markers, an AutoCAD color index (<b>integer</b>). See <a href="#">Color Index Colors</a>.</p> <p>Default = 6 (magenta)</p>
<a href="#">PSEUDO_MARKER</a>	<p>Marker shape for pseudo node errors (<b>integer</b>). See <a href="#">Marker Shapes</a>.</p> <p>Default = 2 (triangle)</p>
<a href="#">SHORT_COLOR</a>	<p>Color for short markers, an AutoCAD color index (positive <b>integer</b>). See <a href="#">Color Index Colors</a>.</p> <p>Default = 1 (red)</p>
<a href="#">SHORT_MARKER</a>	<p>Marker shape for short errors (<b>integer</b>). See <a href="#">Marker Shapes</a>.</p> <p>Default = 1 (octagon)</p>
<a href="#">UNDER_COLOR</a>	<p>Color for undershoot markers, an AutoCAD color index</p>

	(integer). See <a href="#">Color Index Colors</a> . Default = 3 (green)
UNDER_MARKER	Marker shape for undershoot errors (integer). See <a href="#">Marker Shapes</a> . Default = 1 (octagon)
WEED_DISTANCE	<b>!! New !!</b> Weeding distance (real). Default = 15.0
WEED_ANGLE	<b>!! New !!</b> Weed distance (real). Default = 4.0
WEED_SUPPLEMENT_DISTANCE	<b>!! New !!</b> Weed supplement distance (real). Default = 100.0
WEED_SUPPLEMENT_BULGE	<b>!! New !!</b> Weed supplement bulge (real). Default = 1.0



## Cleanup Action Variables

### [Topology Configuration Variables](#)

---

Cleanup action variables store properties for cleanup actions, which are assigned to action lists by [tpm\\_cleanactionlistins](#).

Action lists are a feature of cleanup models. Cleanup action variables are a subset of the [configuration variables](#) data structure.

<a href="#">ARC_TYPE</a>	With the Simplify Objects cleanup action (clean group type <b>128</b> ) <i>only</i> , whether to create arcs ( <b>integer</b> ). <b>1</b> = Create arcs (default) <b>0</b> = Do not create arcs For any action other than <b>128</b> , Simplify Objects, <b>ARC_TYPE</b> does not matter.
<a href="#">CLEAN_TOL</a>	Cleanup tolerance (positive <b>real</b> ). Default = <b>0.01</b> <b>CLEAN_TOL</b> affects the following actions only: <b>8</b> = Delete Duplicates <b>1</b> = Erase Short Objects <b>4</b> = Extend Undershoots <b>16</b> = Snap Clustered Nodes <b>64</b> = Erase Dangling Objects <b>128</b> = Simplify Objects With the following actions, the <b>CLEAN_TOL</b> value does not matter: <b>2</b> = Break Crossing Objects <b>32</b> = Dissolve Pseudo Nodes <b>256</b> = Zero Length Objects
<a href="#">CONVERT</a>	With the Extend Undershoots cleanup action (clean group type <b>4</b> ) <i>only</i> , whether to break the target object (insert a new node) where the extended undershoot intersects it ( <b>integer</b> ). <b>1</b> = Break target (default) <b>0</b> = Do not break target For any action other than <b>4</b> , Extend Undershoots, <b>CONVERT</b> does not matter.



## Topology Variables

### [Topology Configuration Variables](#)

---

Topology variables store properties for topologies, which are created by [tpm\\_mntbuild](#) and the topology analysis functions, [tpm\\_ana\[xx\]](#).

Topology variables are a subset of the [configuration variables](#) data structure.

<b>BUILD_TOL</b>	Tolerance for topology building, buffering, and other processes (positive <b>real</b> ). Default = <b>0.01</b>
<b>CNTR_COLOR</b>	Color for new centroids, an AutoCAD color index ( <b>integer</b> ). Negative <b>integer</b> = Current color Default = <b>-1</b> See <a href="#">Color Index Colors</a> .
<b>CNTR_LAYER</b>	Layer name for new centroids ( <b>string</b> ). <b>nil</b> = Current layer Default = ""
<b>CNTR_TYPE</b>	Entity type of new centroids ( <b>integer</b> ). <b>1</b> = Point (default) <b>2</b> = Block
<b>CNTR_BLOCK</b>	Block name for new centroids ( <b>string</b> ). Default = ""
<b>CREATE_CNTR</b>	Flag for centroids generation ( <b>integer</b> ). <b>1</b> = Generate (default) <b>0</b> = Do not generate
<b>CREATE_MARKERS</b>	Whether errors should be marked with persistent markers as drawn by <a href="#">tpm_cleanerrormark</a> ( <b>integer</b> ). <b>0</b> = Do not mark (default) <b>1</b> = Mark <a href="#">Marker shapes</a> are set by <b>MARKER_HEIGHT</b> and the <a href="#">[XX]_MARKER cleanup variables</a> .
<b>CREATE_NODE</b>	Flag for nodes generation ( <b>integer</b> ). <b>1</b> = Generate (default)

	<p>0 = Do not generate</p>
<b>CREATE_VIEW</b>	<p>Whether errors should be marked with temporary markers as drawn by <a href="#">tpm_cleanerrordraw</a></p> <p>0 = Do not mark 1 = Mark (default)</p>
<b>DEF_OFFSET</b>	<p>Default offset (<b>real</b>).</p> <p>Use when offset value cannot be calculated for an object, for example, if the object lacks necessary <a href="#">object data</a>.</p> <p>Default = <b>1.0</b></p>
<a href="#">DUPLICATE_CENTROID_COLOR</a>	<p>Marker color for duplicate centroid errors, an AutoCAD color index (<b>integer</b>).</p> <p>Default = <b>1</b> (red)</p> <p>See <a href="#">Color Index Colors</a>.</p>
<a href="#">DUPLICATE_CENTROID_MARKER</a>	<p>Marker shape for duplicate centroid errors (<b>integer</b>). See <a href="#">Marker Shapes</a>.</p> <p>Default = <b>4</b> (square)</p>
<b>IGNORE_INCOMPLETE_AREA</b>	<p>What to do if links are encountered that do not belong to any polygon (<b>integer</b>).</p> <p>1 = Ignore them 0 = Cancel topology creation process (default)</p>
<a href="#">INCOMPLETE_AREA_COLOR</a>	<p>Marker color for incomplete area errors, an AutoCAD color index (<b>integer</b>).</p> <p>Default = <b>2</b> (yellow)</p> <p>See <a href="#">Color Index Colors</a>.</p>
<a href="#">INCOMPLETE_AREA_MARKER</a>	<p>Marker shape for incomplete area errors (<b>integer</b>). See <a href="#">Marker Shapes</a>.</p> <p>Default = <b>2</b> (triangle)</p>
<a href="#">INTERSECTION_COLOR</a>	<p>Marker color for intersection errors, an AutoCAD color index (<b>integer</b>).</p> <p>Default = <b>3</b> (green)</p> <p>See <a href="#">Color Index Colors</a>.</p>
<a href="#">INTERSECTION_MARKER</a>	<p>Marker shape for intersection errors (<b>integer</b>). See <a href="#">Marker Shapes</a>.</p> <p>Default = <b>1</b> (octagon)</p>
<a href="#">MISSING_CENTROID_COLOR</a>	<p>Marker color for missing centroid errors, an AutoCAD color index (<b>integer</b>).</p> <p>Default = <b>4</b> (cyan)</p> <p>See <a href="#">Color Index Colors</a>.</p>

<a href="#"><u>MISSING_CENTROID_MARKER</u></a>	Marker shape for missing centroid errors ( <b>integer</b> ). See <a href="#">Marker Shapes</a> . Default = <b>3</b> (rhombus)
<b>NODE_BLOCK</b>	Block name for new nodes ( <b>string</b> ). Default = ""
<b>NODE_COLOR</b>	Color for new nodes, an AutoCAD color index ( <b>integer</b> ): Negative <b>integer</b> = The current color Default = <b>-1</b> See <a href="#">Color Index Colors</a> .
<b>NODE_LAYER</b>	Layer name for new nodes ( <b>string</b> ): <b>nil</b> The current layer Default = ""
<b>NODE_TYPE</b>	Entity type of new nodes ( <b>integer</b> ): <b>1</b> Point (default) <b>2</b> Block
<b>STOP_AT_MISSING_CNTR</b>	What to do if a polygon has no centroid ( <b>integer</b> ): <b>0</b> Create a centroid (default) <b>1</b> Cancel topology creation process
<b>STOP_AT_MULTIPLE_CNTR</b>	What to do if a polygon has more than one centroid ( <b>integer</b> ): <b>0</b> Designate one centroid <b>1</b> Cancel topology creation process (default)



## tpm\_varalloc

[Variables](#) [Functions](#)

---

Allocates a set of configuration variables.

```
ade_id  
tpm_varalloc();
```

Returns a configuration variables ID or `ADE_NULLID`.

Configuration variables are composed of cleanup variables, cleanup action variables, and topology variables. The variables are initialized to their default values. For a list of these variables and their default values, see [Configuration Variables](#).

You can allocate more than one set of configuration variables.



## tpm\_varfree

[Variables Functions](#)

---

Frees a set of configuration variables.

int

```
tpm_varfree(  
    ade_id var_id);
```

Returns **RTNORM** or an error code.

**var\_id** Configuration variables ID.

Configuration variables are composed of cleanup variables, cleanup action variables, and topology variables. The variables are initialized to their default values. For a list of these variables and their default values, see [Configuration Variables](#).



## tpm\_varget

[Variables Functions](#)

---

Gets the value of a configuration variable.

```
int
tpm_varget(
    ade_id var_id,
    char *var_name,
    struct *res);
```

Returns **RTNORM** or an error code.

<code>var_id</code>	Configuration variables ID.
<code>var_name</code>	Variable name.
<code>res</code>	Variable value.

Configuration variables are composed of cleanup variables, cleanup action variables and topology variables. The variables are initialized to their default values. For a list of these variables and their default values, see [Configuration Variables](#).

### [Assigning And Reading Resbuf Values](#)

The following example demonstrates reading and writing a **resbuf** using `tpm_varget` and [tpm\\_varset](#):

```
struct resbuf rb;
ade_id var_id = ADE_NULLID;
int ret = 0;

// Correct ID
var_id = tpm_varalloc();

// Set real value
rb.restype = RTREAL;
rb.resval.rreal = 0.5;
rb.rbnext = NULL;
```

```

ret = tpm_varset(var_id, "CLEAN_TOL", &rb; );

// Get real value
// rb.restype = RTREAL
// rb.resval.rreal = 0.5
ret = tpm_varget(var_id, "CLEAN_TOL", &rb;);

// Set string value
rb.restype = RTSTR;
rb.resval.rstring = (char *)malloc((unsigned)(strlen("LAYER_1") + 1));

ret = tpm_varset(var_id, "LINK_LAYER", &rb; );

// Get string value
// rb.restype = RTSTR
// rb.resval.rstring = "LAYER_1"
ret = tpm_varget(var_id, "LINK_LAYER", &rb; );

if ( ret == RTNORM )
free (rb.resval.rstring);

```

### Sample Code

The following sample retrieves the value for the NODE\_LAYER configuration variable using `tpm_varget()`. The topology containing the desired information is loaded then opened using `tpm_acload()` and `tpm_acopen()` respectively. A set of configuration variables is allocated using `tpm_varalloc()`, then filled using `tpm_infobuildvar()`. The topology can now be closed and unloaded using `tpm_acclose` and `tpm_acunload` respectively. A `resbuf` is created which will be populated with the configuration variable value using `tpm_varget()`, which is called with all required parameters. The return code is validated against `RTNORM` and appropriate status information is displayed. The `resbuf` is then released as required.

```

char* pszTopoName = "netTopo";
int topoWriteAccess = 0;
int returnCode = tpm_acload(pszTopoName, 0);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
ade_id varId = tpm_varalloc();
returnCode = tpm_infobuildvar(topoId, varId);
tpm_acclose(topoId);
char* pszConfigVarName = "NODE_LAYER";
struct resbuf* pConfigVarsRb = acutBuildList(RTSTR, "", 0);
returnCode = tpm_varget(varId, pszConfigVarName, pConfigVarsRb);

```

```
if (RTNORM == returnCode){
    acutPrintf(
        "\n\nThe \"%s\" configuration variables value is: \"%s\"."
        , pszConfigVarName, pConfigVarsRb->resval.rstring);
}
else {
    acutPrintf(
        "\n\nThe specified configuration variable was not retrieved.");
}
acutRelRb(pConfigVarsRb);
```



## tpm\_varset

[Variables Functions](#)

---

Sets the value of a configuration variable.

```
int
tpm_varset(
    ade_id var_id,
    char *var_name,
    struct resbuf *valres);
```

Returns **RTNORM** or an error code.

<code>var_id</code>	Configuration variables ID.
<code>var_name</code>	Variable name.
<code>valres</code>	Variable value contained in a <b>resbuf</b> .

Configuration variables are composed of cleanup variables, cleanup action variables, and topology variables. The variables are initialized to their default values. For a list of these variables and their default values, see [Configuration Variables](#).

Examples of building a **resbuf** for the **valres** argument.

```
rb = acutBuildList(RTSHORT, 4, 0); // integer value
rb = acutBuildList(RTREAL, 1.0, 0); // real value
rb = acutBuildList(RTSTR, "0", 0); // string value
```

You must release the **resbuf**.

The following sample allocates a set of configuration variables using **tpm\_varalloc()**. A **resbuf** is created containing the layer name that will be associated with the **LINK\_LAYER** configuration variable.

**Tpm\_varset()** is called with all required parameters and the return code is evaluated for the appropriate display message. The **resbuf** is then released as required.

```
ade_id varId = tpm_varalloc();
char* pszConfigVarName = "LINK_LAYER";
struct resbuf* pConfigVarsRb = acutBuildList(RTSTR, "Links", 0);
```

```
int returnCode = tpm_varset(varId, pszConfigVarName, pConfigVarsRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe specified configuration variable has been set.");
}
else {
    acutPrintf(
        "\nThe specified configuration variable was not set.");
}
acutRelRb(pConfigVarsRb);
```



## tpm\_cleanactionlistgetat

### [Cleanup Functions](#)

---

Gets the cleanup action at a given list position.

```
int
*tpm_cleanactionlistgetat
    ade_id clean_var_id,
    long index,
    ade_id action_var_id);
```

Returns a cleanup action as a clean group type. See [tpm\\_cleangrouptype](#) for a list of types.

`clean_var_id`      The cleanup variables ID returned by [tpm\\_varalloc](#).

`index`            The list position to access.

`action_var_id`    The cleanup action variables ID returned by [tpm\\_varalloc](#).

The `clean_var_id` argument references properties for the cleanup operation that you are preparing to initiate (see [Cleanup Variables](#)). These properties include the action list.

The `index` argument is a zero-based position in the action list. A value greater than or equal to the list size or less than 0 returns an error.

The `action_var_id` argument references properties affecting the specific cleanup action that you are getting (see [Cleanup Action Variables](#)). Use [tpm\\_varget](#) or [tpm\\_varlist](#) to read them after calling `tpm_cleanactionlistgetat`.



## tpm\_infobuildvar

### [Topology Information Functions](#)

---

Gets the configuration values of a topology.

```
int  
tpm_infobuildvar(  
    ade_id tpm_id,  
    ade_id var_id);
```

Returns **RTNORM** or an error code.

**tpm\_id**      Topology ID.  
**var\_id**      Topology variables ID.

The topology variables ID references a set of [topology variables](#) in which to store the values that this function gets.

If no topology variables are allocated, call [tpm\\_varalloc](#) to allocate a set of them and return their ID.

To read the values that this function gets, use [tpm\\_varget](#) or [tpm\\_varlist](#). To build a new topology using these variables, use [tpm\\_mntbuild](#).

The following sample opens a topology, reads its configuration values using `tpm_infobuildvar()`, and gets its node layer. The topology is then closed, unloaded and the **resbuf** is released as required.

```
char* pszTopoName = "I-95_Corridor";  
int resultCode = tpm_acload(pszTopoName, 0);  
ade_id topoId = tpm_acopen(pszTopoName, 0);  
ade_id topoConfigVarId = tpm_varalloc();  
resultCode = tpm_infobuildvar(  
    topoId,  
    topoConfigVarId);  
  
char* pszConfigVarName = "NODE_LAYER";  
struct resbuf* pConfigVarsRb = acutBuildList(RTSTR, "", 0);  
resultCode = tpm_varget(  
    topoConfigVarId,
```

```
        pszConfigVarName,
        pConfigVarsRb);
if (RTNORM == resultCode){
    acutPrintf(
        "\nThe %s variable in the %s topology contains the value \"%s\"."
        , pszConfigVarName, pszTopoName, pConfigVarsRb->resval.rstring);
}
else {
    acutPrintf(
        "\nCould not determine the %s variable value."
        , pszConfigVarName);
}
resultCode = tpm_acclose(topoId);
resultCode = tpm_acunload(pszTopoName);
acutRelRb(pConfigVarsRb);
```



## tpm\_varlist

[Variables Functions](#)

---

Gets all the values in a set of configuration variables.

```
struct resbuf
*tpm_varlist(
    ade_id var_id);
```

Returns a **resbuf** of name-value pairs or **NULL**.

**var\_id** Configuration variables ID.

Configuration variables are composed of cleanup variables, cleanup action variables, and topology variables. The variables are initialized to their default values. For a list of these variables and their default values, see [Configuration Variables](#).

The calling function must release the **resbuf**.

Each **resbuf** has this format:

```
variable name, (RTSTR) . value, (RTSTR, RTREAL, RTSHORT)
```

The following sample allocates a set of configuration variables using **tpm\_varalloc()**, then populates a **resbuf** with name-value pairs associated with the current configuration variable set using **tpm\_varlist()**. The contents of the **resbuf** are displayed, then the **resbuf** is released as required.

```
ade_id varId = tpm_varalloc();
struct resbuf* pConfigVarsRb = tpm_varlist(varId);
if (NULL != pConfigVarsRb){
    struct resbuf* rb = pConfigVarsRb;
    while(NULL != rb) {
        if (rb->restype == RTSTR) {
            acutPrintf(
                "\nThe \"%s\" property contained the value:"
                , rb->resval.rstring);
            if (NULL != (rb = rb->rbnext)) {
                switch(rb->restype)
```

```
{
    case RTSTR:
        acutPrintf(
            "\"%s\""
            , rb->resval.rstring);
        break;
    case RTREAL:
        acutPrintf(
            "%.2lf"
            , rb->resval.rreal);
        break;
    case RTSHORT:
        acutPrintf(
            "%d"
            , rb->resval.rint);
        break;
    default:
        break;
}
}
}
rb = rb->rbnext;
}
}
else {
    acutPrintf(
        "\nThe configuration variable list could not be retrieved.");
}
acutRelRb(pConfigVarsRb);
```

# Object Model Legend

The arrow buttons are for navigation.

-  Click to navigate down the tree.
-  Click to navigate up the tree (toward AcMapSession).

The chevron button links to more information.

-  Click to pop up a note.

Not all classes are members of AutoCAD Map.

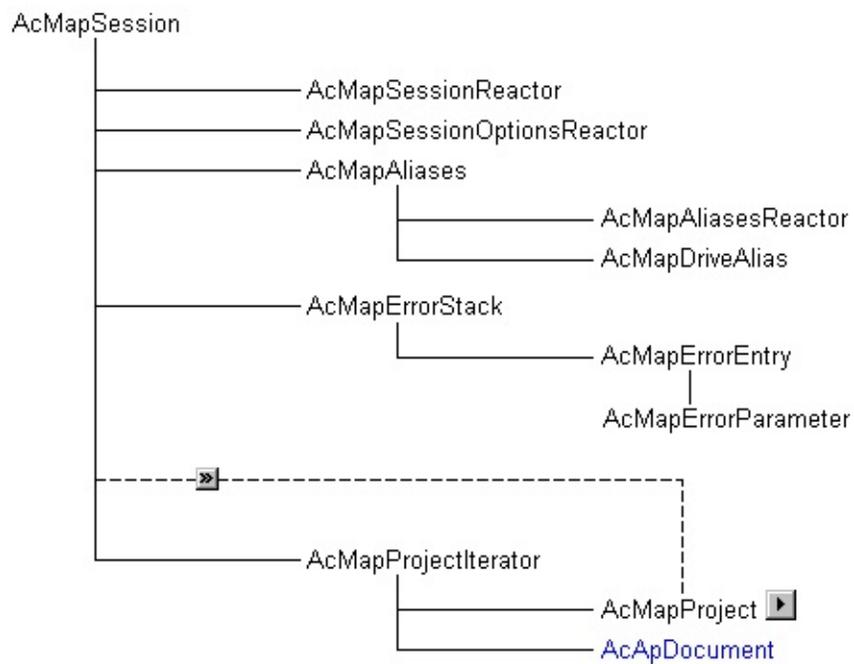
**NamesInBlack** AutoCAD Map ObjectARX classes. Click a class name to open its Help topic.

**NamesInBlue** AutoCAD ObjectARX classes. Refer to AutoCAD ObjectARX Help.

Containment is managed by functions of the containing classes. For example, AcMapSession contains an AcMapAliases object through the AcMapSession::GetAliases function, and AcMapAliases in turn contains one or more AcMapDriveAlias objects through the AcMapAliases::GetAlias function. AcMapAliases has other functions as well for managing containment: functions to add, count, remove, and so on.

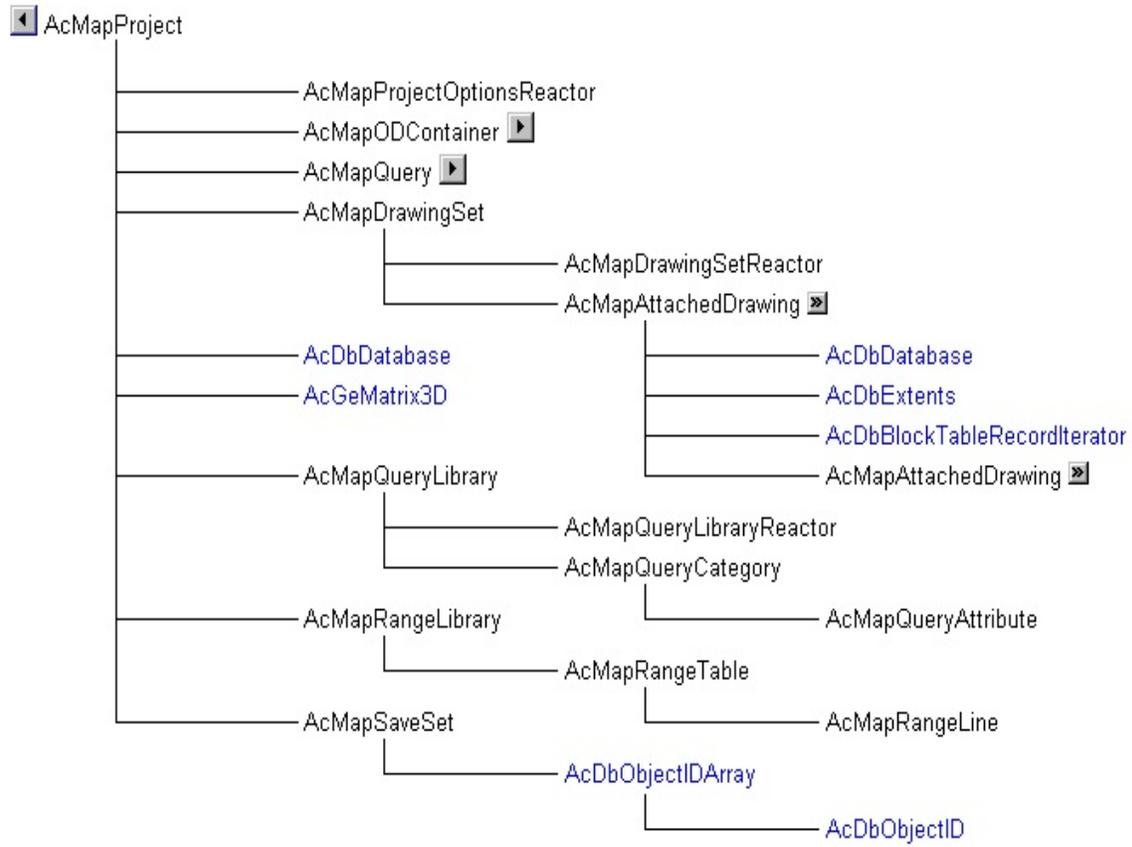
# Object Model: AcMapSession

[Inheritance](#) [Legend](#)



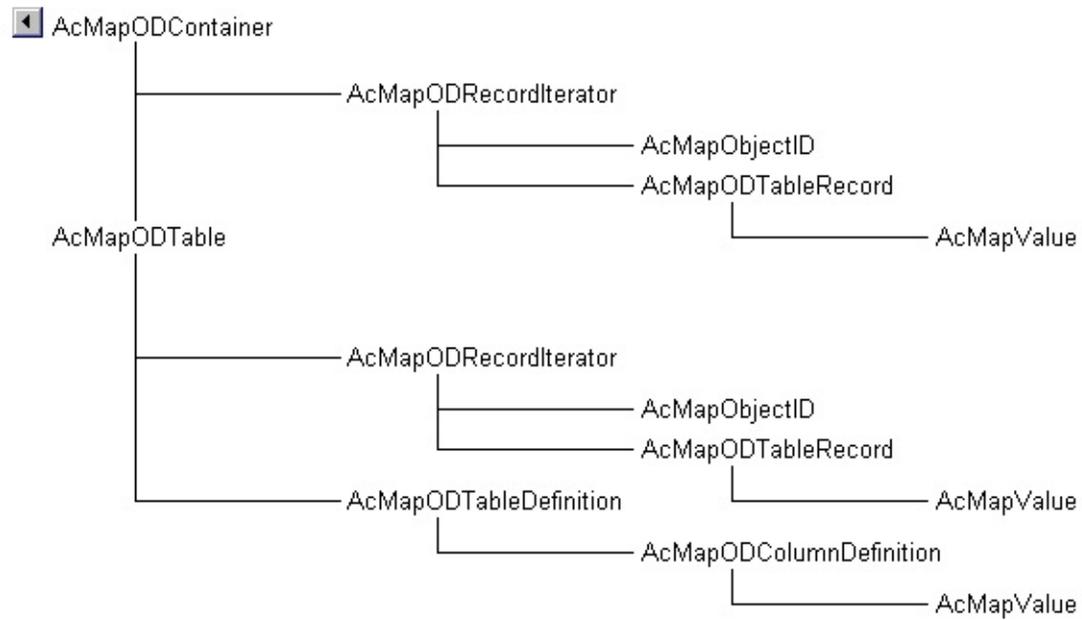
# Object Model: AcMapProject

[Inheritance](#) [Legend](#)



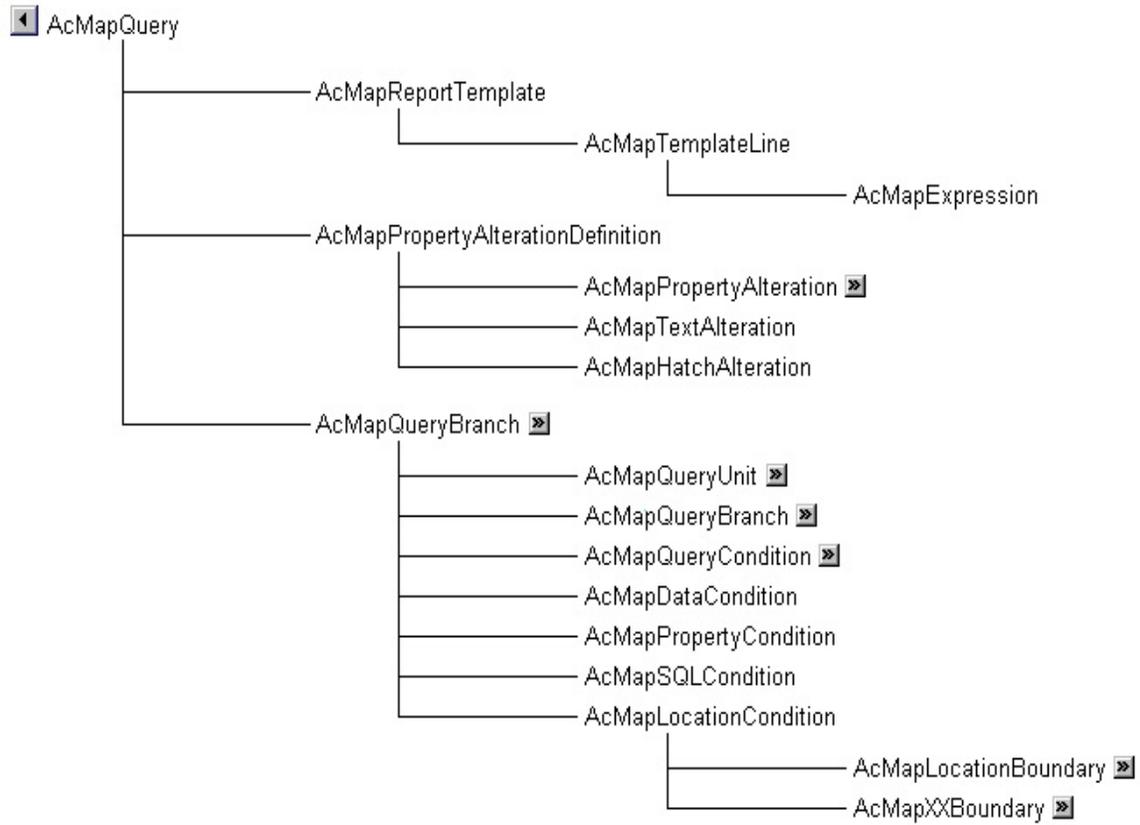
# Object Model: AcMapODContainer

[Inheritance](#) [Legend](#)



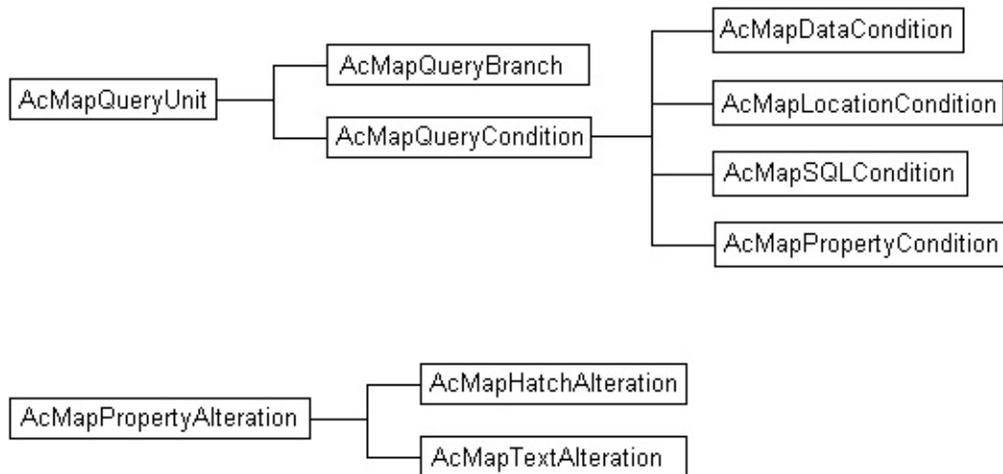
# Object Model: AcMapQuery

[Inheritance](#) [Legend](#)



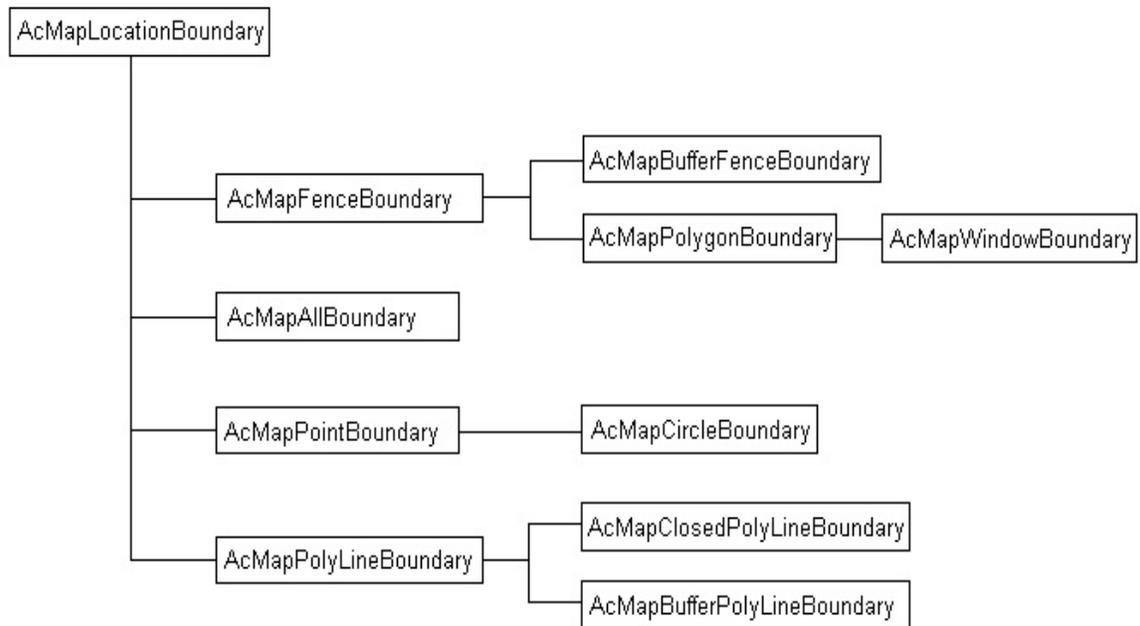
# Inheritance: Query Classes

## [Object Model](#)



# Inheritance: Boundary Classes

## Object Model



# AcMapSession class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapSession.h.

An instance of the AcMapSession class represents a Map session, which is a single instance of AutoCADMap that is currently running.

An AcMapSession contains the session's error stack (AcMapErrorStack class), its aliases (AcMapAliases class), its projects (AcMapProject class) and its project iterator (AcMapProjectIterator class).

Do not subclass from this class.

To get an AcMapGetSession pointer, call AcMapGetSession, which is declared an extern C function in MapSession.h. AutoCAD Map releases the memory allocated for AcMapGetSession, so do not attempt to delete it yourself.

The following example implements a command that declares a pointer to an AutoCAD Map session, and immediately sets the pointer to NULL. The example calls AcMapGetSession to get the AcMapSession pointer. For simplicity, the code does not show routine error checking.

```
#include "StdAfx.h"
#include "StdArx.h"
#include <MapArxApi.h>

void asdkhelpcreatesession()
{
    AcMapSession *mapApi = NULL;
    mapApi = AcMapGetSession();
    if (mapApi == NULL) {
        acutPrintf ("\nCan't connect to MAP");
        return;}
    else{
        acutPrintf("\nCreated AcMapSession");}
    acedRetVal();
}
```

# AcMapSessionOptionsReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapSessionOptionsReactor class is used to represent a session options reactor, which notifies an application when a session option is modified.

To create a specific project options reactor, subclass from AcMapSessionOptionsReactor.

Use AcMapSession::AddOptionsReactor to add an options reactor to a session and AcMapSession::RemoveOptionsReactor to remove it.

You use the new operator to get an AcMySessionOptionsReactor pointer, and call AcMapSession::RemoveOptionsReactor to deallocate memory, as shown in the following example. For simplicity, routine error checking is not shown.

```
class AcMySessionOptionsReactor : public AcMapSessionOptionsReactor
{
    // See MapDemoApp.cpp distributed with AutoCAD Map ObjectARX.
};

static AcMySessionOptionsReactor *pSysOptReactor ;

AcRx::AppRetCode acrxEntryPoint(AcRx::AppMsgCode msg, void *appId)
{
    AcMapSession *mapApi = NULL;
    switch (msg)
    {
        case AcRx::kInitAppMsg:
            acrxRegisterAppMDIAware(appId);
            //register commands -- code not shown
            pSysOptReactor = new AcMySessionOptionsReactor ();
            mapApi = AcMapGetSession();
            if (mapApi)
                mapApi->AddOptionsReactor(pSysOptReactor) ;
            break ;
        case AcRx::kUnloadAppMsg:
            //remove commands -- code not shown
            break ;
        case AcRx::kUnloadDwgMsg:
            mapApi = AcMapGetSession();
            if (mapApi)
            {
                AcMapProject *pProj ;
                if (mapApi->GetProject(pProj))
                    mapApi->RemoveOptionsReactor(pSysOptReactor) ;
            }
            break ;
    }
    return AcRx::kRetOK;
}
```



# AcMapSessionReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapSessionReactor class is used to represent a session reactor, which notifies an application of session events.

To create a specific session reactor, subclass from AcMapSessionReactor.

Use AcMapSession::AddSessionReactor to add a session reactor to a session and AcMapSession::RemoveSessionReactor to remove it. Delete the pointer to the reactor using the delete operator when you unload the application.

# AcMapProject class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapProj.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapProject class, a project, is the principal container for objects used in AutoCAD Map. Within a Map session, a user can open and close multiple projects.

The project contains objects of the following classes:

- AcMapDrawingSet class
- AcMapRangeLibrary class
- AcMapQueryLibrary class
- AcMapSaveSet class
- AcMapODContainer class
- AcMapExpression class
- AcMapQuery class

You cannot instantiate a project programmatically. A project is instantiated when a Map user opens a document. You can get a pointer to a project with the AcMapSession::GetProject function or with the Project functions of various classes. Do not delete memory associate with the project pointer.

# AcMapProjectIterator class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapProj.h.

An instance of the AcMapProjectIterator class represents a project iterator, which is used to iterate through the projects in a session.

Do not subclass from this class.

You get a AcMapProjectIterator pointer by calling AcMapSession::GetProjectIterator. You are responsible for deleting memory allocated for the iterator using the delete operator.

The following code fragment shows how to create and delete a project iterator. For simplicity, routine error checking is not shown.

```
AcMapSession *mapApi = NULL;
mapApi = AcMapGetSession();
cMapProjectIterator *pIterator = mapApi->GetProjectIterator();
AcMapProject *pProject ;
for (; !pIterator->Done(); pIterator->Next())
{
    pIterator->GetCurrentObject(pProject) ;
    . . .
}
delete pIterator ;
```

# AcMapProjectOptionsReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapProjectOptionsReactor class is used to represent a project options reactor, which notifies an application when a project option is modified.

To create a specific project options reactor, subclass from AcMapProjectOptionsReactor.

Use AcMapProject::AddOptionsReactor to add an options reactor to a project and AcMapProject::RemoveOptionsReactor to remove it.

For simplicity, the following example does not show routine error checking.

```
class AcMyProjectOptionsReactor : public AcMapProjectOptionsReactor
{
    // See MapDemoApp.cpp distributed with AutoCAD Map ObjectARX.
};

static AcMyProjectOptionsReactor *pPrefReact;

AcRx::AppRetCode acrxEntryPoint(AcRx::AppMsgCode msg, void *appId)
{
    AcMapSession *mapApi = NULL ;
    switch (msg)
    {
        case AcRx::kInitAppMsg:
            acrxRegisterAppMDIAware(appId);
            //register commands -- code not shown
            pPrefReact = new AcMyProjectOptionsReactor();
            mapApi = AcMapGetSession();
            if (mapApi)
                pProj->AddOptionsReactor(pPrefReact) ;
            break ;
        case AcRx::kUnloadAppMsg:
            //remove commands -- code not shown
            break ;
        case AcRx::kUnloadDwgMsg:
            mapApi = AcMapGetSession();
            if (mapApi)
            {
                AcMapProject *pProj ;
                if (mapApi->GetProject(pProj))
                    pProj->RemoveOptionsReactor(pPrefReact);
            }
            break ;
    }
    return AcRx::kRetOK;
}
```



# AcMapAliases class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

The AcMapAliases class represents a list of drive aliases that have been defined for the session. It provides support for the AcMapDriveAlias class, whose instances represent the individual drive aliases used in an AutoCAD Map session.

You must create a drive alias for a drawing before it can be attached.

Do not subclass from this class or delete the AcMapAliases pointer.

You get an pointer to the session's aliases by calling AcMapSession::GetAliases as shown in the following example, which implements a command called accessaliases. For simplicity, the code does not show routine error checking.

```
#include "StdAfx.h"
#include "StdArx.h"
#include <MapArxApi.h>

void asdkhelpaccessaliases()
{
    AcMapSession *mapApi = NULL;
    AcMapAliases *pAliases = NULL;

    mapApi = AcMapGetSession();
    mapApi->GetAliases(pAliases);
    // Use pAliases to access functions of AcMapAliases -- code not shown
    acedRetVal();
}
```

# AcMapAliasesReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapAliasesReactor class is used to represent a drive alias reactor, which notifies an application of drive alias events.

To create a specific drive aliases reactor, subclass from AcMapAliasesReactor.

Use AcMapAliases::AddReactor to add the reactor to a drive alias and AcMapAliases::RemoveReactor to remove it. Delete the pointer to the reactor using the delete operator when you unload the application, as shown in the following example. For simplicity, error checking is not shown.

```
class AcMyAliasesReactor : public AcMapAliasesReactor
{
    // See MapDemoApp.cpp distributed with AutoCAD Map ObjectARX.
};

static AcMyAliasesReactor *pAliasReact ;

AcRx::AppRetCode acrxEntryPoint(AcRx::AppMsgCode msg, void *appId)
{
    AcMapSession *mapApi = NULL;
    switch (msg) {
        case AcRx::kInitAppMsg:
            acrxRegisterAppMDIAware(appId);
            //register commands -- code not shown
            pAliasReact = new AcMyAliasesReactor() ;
            mapApi = AcMapGetSession();
            AcMapAliases *pAlias = NULL ;
            mapApi->GetAliases(pAlias);
            pAlias->AddReactor(pAliasReact) ;
            break ;
        case AcRx::kUnloadAppMsg:
            delete pAliasReact ;
            pAliasReact = NULL ;
            break ;
        case AcRx::kUnloadDwgMsg:
            mapApi = AcMapGetSession();
            AcMapAliases *pAlias = NULL ;
            mapApi->GetAliases(pAlias);
            pAlias->RemoveReactor(pAliasReact) ;
            break ;
    }
    return AcRx::kRetOK;
}
```



# AcMapDriveAlias class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

An instance of the AcMapDriveAlias class represents a drive alias used in an AutoCAD Map session.

A drive alias allows you to assign a directory or network path string to a single AutoCAD name, such as a single drive letter, and to substitute the alias name for the directory path string.

The AcMapAliases class contains the AcMapDriveAlias class. You create an AcMapDriveAlias object using the AcMapAliases::AddAlias and FindAlias function. You are responsible for deleting memory allocated for the AcMapDriveAlias using the delete operator.

Do not subclass from this class.

# AcMapAttachedDrawing class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An attached drawing, also called a source drawing.

An attached drawing can contain other attached drawings nested within it.

Do not subclass from this class. You create an attached drawing using the `AcMapDrawingSet::GetDrawing` or `AcMapDrawingSet::AttachDrawing`. These functions return a pointer by reference to the attached drawing that you are responsible for deleting using the delete operator, as shown in the example.

The following example creates a command that gets a pointer to an attached drawing called, demo.dwg. Delete the memory allocated for the attached drawing, using the delete operator as shown here. The example assumes that you already attached the drawing and its location is specified by ALIAS. For simplicity, routine error checking is not shown.

```
#include "StdAfx.h"
#include "StdArx.h"
#include <MapArxApi.h>
#include <MapProj.h>
#include <MapStringArray.h>

void asdkhelpgetdwg()
{
    AcMapAttachedDrawing *pDwg = NULL;
    AcMapSession *mapApi = NULL;
    AcMapProject *pProj = NULL;
    AcMapDrawingSet *pDSet = NULL;

    mapApi = AcMapGetSession();
    mapApi->GetProject(pProj);
    pProj->GetDrawingSet(pDSet);
    pDSet->GetDrawing(pDwg, "ALIAS:\\demo.dwg", Adesk::kTrue);
    // do something with pDwg
    delete pDwg;
    acedRetVoid();
}
```



# AcMapDrawingSet class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

The attached drawings of a project. A project contains a single drawing set.

A drawing set is a tree. Some drawings are attached directly to the drawing set, and each attached drawing can have other attached drawings nested within it.

To alter a drawing set, a user must have the Alter Drawing Set user privilege. Set this privilege in the AutoCAD Map User Administration dialog.

Do not subclass from this class.

To get an AcMapDrawingSet pointer, call AcMapProject::GetDrawingSet, as shown in the following example. Do not attempt to delete the memory associated with the drawing set pointer. For simplicity, the example does not show routine error checking.

```
AcMapSession    *mapApi = NULL;
AcMapProject    *pProj = NULL;
AcMapDrawingSet *pDSet = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetDrawingSet(pDSet);
pDSet->ZoomExtents();
```

# AcMapDrawingSetReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapDrawingSetReactor class is used to represent a drawing set reactor, which supplies callbacks to notify an application immediately of drawing set events.

To create a specific drawing set reactor, subclass from AcMapDrawingSetReactor.

Use AcMapDrawingSet::AddReactor to add a reactor to a drawing set and AcMapDrawingSet::RemoveReactor to remove it.

Use AcMapDrawingSet::AddReactor to add the reactor to a drawing set and AcMapDrawingSet::RemoveReactor to remove it. Delete the pointer to the reactor using the delete operator when you unload the application, as shown in the following example. For simplicity, error checking is not shown.

```
class AcMyDrawingSetReactor : public AcMapDrawingSetReactor
{
    // See MapDemoApp.cpp distributed with AutoCAD Map ObjectARX.
};

static AcMyDrawingSetReactor *pDSetReact ;

AcRx::AppRetCode acrxEntryPoint(AcRx::AppMsgCode msg, void *appId)
{
    AcMapSession *mapApi = NULL;
    switch (msg) {
        case AcRx::kInitAppMsg:
            acrxRegisterAppMDIAware(appId);
            //register commands -- code not shown
            pDSetReact = new AcMyDrawingSetReactor() ;
            mapApi = AcMapGetSession();
            AcMapDrawingSet *pDSet = NULL ;
            pProj->GetDrawingSet(pDSet);
            pDSet->AddReactor(pDSetReact) ;
            break ;
        case AcRx::kUnloadAppMsg:
            delete pDSetReact ;
            pDSetReact = NULL ;
            break ;
        case AcRx::kUnloadDwgMsg:
            mapApi = AcMapGetSession();
            AcMapDrawingSet *pDSet = NULL ;
            pProj->GetDrawingSet(pDSet);
            pDSet->RemoveReactor(pDSetReact) ;
            break ;
    }
    return AcRx::kRetOK;
```



# AcMapObjectId class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapObjectId.h.

The AcMapObjectId class implements an AutoCAD Map database identifier. This class provides identifiers for AutoCAD entities which are persistent across database load and unload operations.

An AutoCAD Map database identifier consists of an object handle and the identifier of the source drawing to which the object belongs.

Do not subclass from this class.

# AcMapExpression class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapExpression class is an expression, which you can use throughout an AutoCAD Map project. You use AcMapExpression to define, set, get, and execute an expression or manipulate template lines. An expression is a combination of operands and operators that evaluates to a single value. The same expression always evaluates to the same value. For more information about expressions, click Expression Evaluator on the Contents tab of AutoCAD Map Help.

Do not subclass from this class.

You call AcMapProject::DefineExpression to get an AcMapExpression pointer.. You are responsible for deleting the memory allocated for the object using the delete operator, as shown in the following example. For simplicity, the code does not show routine error checking.

```
char buff[132] ;
AcMapSession *mapApi = NULL ;
AcMapProject *pProj = NULL ;
AcMapExpression *pExpress = NULL ;
AcDbObjectId Id = 0 ;

AcMapValue Val ;
mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
*buff = EOS ;
acedGetString (1, "Enter an expression: ", buff);
pProj->DefineExpression(pExpress, buff) ;
// Use the expression -- code not shown
delete pExpress ;
```

# AcMapODColumnDefinition class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapODColumn.h.

The AcMapODColumnDefinition class is used to create a column definition, which can then be passed to AcMapODTableDefinition::AddColumn to build a table definition. The column definition contains the column's data type, name, optional description, and default value.

Do not subclass from this class.

To create a column definition, you instantiate this class using the new operator and release memory allocated for it using the delete operator, as shown in the following example. For simplicity, routine error checking is not shown.

```
AcMapSession *mapApi = NULL;
AcMapProject *pProj = NULL;
AcMapODContainer *pODCont = NULL;
AcMapODTableDefinition *pTabDef = NULL;
AcMapODColumnDefinition *pColDef1 = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetODContainer(pODCont);
pTabDef = new AcMapODTableDefinition();
pColDef1 = new AcMapODColumnDefinition();
pColDef1->SetName("Residential");
pColDef1->SetDescription("Residential R1-R3");
pColDef1->SetType(AcMap::kCharacter);
pColDef1->SetDefaultValue("R1");
pTabDef->AddColumn(*pColDef1);
pODCont->CreateODTable("MinimalTable", *pTabDef, "Simple table",
    Adesk::kFalse);
if (pColDef1) delete pColDef1;
if (pTabDef) delete pTabDef;
```

# AcMapODContainer class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapArxApi.h.

Object data about a source drawing is stored in object data tables that are part of the drawing. The ODContainer class serves as a container for object data tables and their components.

The AutoCAD Map ObjectARX API creates an instance of AcMapODContainer when you open a project. You get a pointer to this instance by calling AcMapProject::GetODContainer to access the project's object data tables. Do not attempt to delete this pointer.

Do not subclass from this class.

The following code fragment gets a pointer to the instance of AcMapODContainer and accesses the project object data tables with it. For simplicity, routine error checking is not shown.

```
AcMapSession *mapApi = NULL;
AcMapProject *pProj = NULL;
AcMapODContainer *pODCont = NULL;
AcMapODTable *pODTable = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetODContainer(pODCont);
pODCont->GetODTable(pODTable, "Zones");
```

# AcMapODRecordIterator class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapODIterator.h.

An instance of the AcMapODRecordIterator class, an object record iterator, is used to iterate through the records in an object data table.

Do not subclass from this class.

# AcMapODTable class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapODTable.h file.

The AcMapODTable class is used to manage object data tables.

Do not subclass from this class.

# AcMapODTableDefinition class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapODDefinition.h.

The AcMapODTableDefinition class is used to create a table definition, which can then be passed to AcMapODContainer::CreateODTable to build an object data table. The table definition defines the object data table's columns. You must first add columns to the table definition creating the table.

Do not subclass from this class.

To create a table definition, instantiate this class using the new operator and delete memory allocated for it using the delete operator, as shown in the following example. For simplicity, routine error checking is not shown.

```
AcMapSession *mapApi = NULL;
AcMapProject *pProj = NULL;
AcMapODContainer *pODCont = NULL;
AcMapODTableDefinition *pTabDef = NULL;
AcMapODColumnDefinition *pColDef1 = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetODContainer(pODCont);
pTabDef = new AcMapODTableDefinition();
pColDef1 = new AcMapODColumnDefinition();
pColDef1->SetName("Residential");
pColDef1->SetDescription("Residential R1-R3");
pColDef1->SetType(AcMap::kCharacter);
pColDef1->SetDefaultValue("R1");
pTabDef->AddColumn(*pColDef1);
pODCont->CreateODTable("MinimalTable", *pTabDef, "Simple table",
Adesk::kFalse);
if (pColDef1) delete pColDef1;
if (pTabDef) delete pTabDef;
```

# AcMapODTableRecord class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapODRecord.h.

An instance of the AcMapODTableRecord class represents a record in an object data table.

Do not subclass from this class.

# AcMapValue class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapValue.h.

An instance of the AcMapValue class represents a value in an object data table record.

The value can be integer, double, char \*, or point. The value's initial data type is set when the first value is assigned to it. The AcMapValue class has functions for explicit data type casting.

The number and data types of AcMapValue objects that are contained by an AcMapODTableRecord object must correspond with the number and data types of AcMapODColumnDefinition objects that are contained by the corresponding AcMapODTableDefinition object.

# AcMapQuery class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapQuery class represents a defined query.

Do not subclass from this class.

## Creating a Query

The AcMapQuery class does not have a constructor, and you cannot create and modify its objects directly. After you have created the query definition, instantiate an AcMapQuery object using the AcMapProject::CreateQuery function. This creates the query object in the project, making the query available to the application. You are responsible for deleting the query, as shown in the following example.

```
// For simplicity, error checking is not shown.

#include "StdAfx.h"
#include "StdArx.h"
#include <MapArxApi.h>
#include <MapProj.h>
#include <MapQuery.h>

AcMapSession *mapApi = NULL;
AcMapProject *pProj = NULL;
AcMapQuery *pQuery = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->CreateQuery(pQuery, Adesk::kTrue);
// define and execute the query -- code not shown
delete pQuery;
```

## Defining a Query

A [BSPSPopupOnMouseOver\(event\);">query definition](#) is built from an AcMapQueryBranch object, which describes the conditions that form the selection criteria of the query. Create the query branch and pass it to the AcMapQuery::Define function to create a query definition. See AcMapQueryBranch class for information about creating the query branch.

## Modifying a Query

To modify an existing query, create a new query branch. Then call AcMapQuery::Define to redefine the query using the new query branch.

## Executing a Query

Before executing a query, you may also want to set the mode (using AcMapQuery::SetMode), enable or disable [BSPSPopupOnMouseOver\(event\);">property alteration](#) (using AcMapQuery::EnablePropertyAlteration) or create a report template (using AcMapQuery::GetReportTemplate) for the query. before executing a query. Then call AcMapQuery::Run to execute the query.

The following example, which assumes drawings are attached, shows how to create a command that creates, defines, and executes a query. For simplicity, error checking is not shown.

```
#include "StdAfx.h"
#include "StdArx.h"
#include <MapArxApi.h>
#include <MapProj.h>
#include <MapQuery.h>
#include <MapBoundary.h>

void asdkhelpqueryall()
{
    AcMapSession    *mapApi = NULL;
    AcMapProject    *pProj = NULL;
    AcMapDrawingSet *pDSet = NULL;
    AcMapQueryBranch *pQBranch = NULL;
    AcMapLocationCondition *pLocationCondition = NULL;
    AcMapQuery *pNewQuery = NULL;

    mapApi = AcMapGetSession();
    mapApi->GetProject(pProj);
    pProj->GetDrawingSet(pDSet);
    pQBranch = new AcMapQueryBranch();
    pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
    pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
    pProj->CreateQuery(pNewQuery, Adesk::kFalse);
    pNewQuery->Clear(Adesk::kTrue);
    pNewQuery->SetMode(AcMap::kQueryDraw);
    pQBranch->AppendOperand(pLocationCondition);
    pNewQuery->Define(pQBranch);
    pNewQuery->Run();
    pDSet->ZoomExtents();
    delete pLocationCondition;
    delete pQBranch;
    delete pNewQuery;
    acedRetVoid();
}
```

## Error Codes

If the return value of a query function is of type `AcMap::EErrorCode` and the function succeeds, it returns `AcMap::kOk`. Error codes denoting unsuccessful conditions are listed under `AcMap::EErrorCode`.

# AcMapQueryAttribute class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapArxApi.h.

The AcMapQueryAttribute class is used to represent a query that has been saved either with a project or in an external file. This is different from the AcMapQuery class, which is used to represent a query in memory.

To create an AcMapQueryAttribute object, pass NULL as the query to AcMapQueryCategory::GetQuery.

Do not subclass from this class.

# AcMapQueryBranch class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

The AcMapQueryBranch class is derived from the AcMapQueryUnit class. It is used by the AcMapQuery::Define to build or modify a query definition.

You get an AcMapQueryBranch pointer using the new operator and you are responsible for deleting the allocated memory using the delete operator.

A query branch is a tree constructed of one or more query units. Each query unit in the query branch can be either a subordinate query branch or an instance of one of the following condition classes:

AcMapLocationCondition, AcMapSQLCondition, AcMapDataCondition, and AcMapPropertyCondition. The conditions, which hang off the query branch, describe the query's selection criteria. Structurally, a condition is a simple node in the query branch.

Do not subclass from this class.

# AcMapQueryCategory class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapArxApi.h.

The AcMapQueryCategory class is used to represent a query category, which is used as a container for saved queries in a query library.

Do not subclass from this class.

# AcMapQueryUnit class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

A query unit represents either a query condition or a subordinate query branch. It is sometimes referred to as a query node or a query operand.

Every query unit contains a `= 4) BSPSPopupOnMouseOver(event)::">join operator` which is used to combine query units logically. The join operator of the first query unit within a query branch is ignored.

Changes to a query unit affect the query only after a call to the `AcMapQuery::Define` function.

The `AcMapQueryBranch` and `AcMapQueryCondition` classes are derived from the `AcMapQueryUnit` class.

Do not subclass from this class.

# AcMapReportTemplate class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapTemplate.h.

The AcMapReportTemplate class is used to represent a report template, which AutoCAD Map uses to format the output of a query in a text file.

A report template contains of one or more column definitions, also called template lines.

Do not subclass from this class.

# AcMapTemplateLine class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapTemplate.h.

The AcMapReportTemplateLine class is used to represent a column definition in a report template. The definition is contained in the column definition expression.

Do not subclass from this class.

# AcMapSaveSet class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

When you edit objects in a project, your changes are saved to the project file (DWG file) by default. You can also use a save set to save changes back to their source drawings. The save set enables users to make changes to their drawings persistent.

The AcMapSaveSet class is used to create a save set, which contains the drawing database identifiers of the drawing objects whose changes are saved back to their source drawing files.

A project has a single save set. Create a save set with the AcMapProject::GetSaveSet function.

Do not subclass from this class.

During an edit session in a network environment, where the administrator has set the object locking preference, AutoCAD Map locks those objects in source drawing files that are specified in a save set so that other users cannot modify the objects.

# AcMapQueryLibrary class

[Functions](#) [Object Model](#) [See Also](#)

This class is defined in MapArxApi.h.

The AcMapQueryLibrary class is used to represent a query library, which AutoCAD Map uses to save queries with a project. The queries can be saved with the project or in an external file.

A query library is created automatically when a project is created. There is only one query library per project.

Within a query library, queries are be organized into an dynamic array of query catagories.

Do not subclass from this class.

# AcMapQueryLibraryReactor class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapReactors.h.

The AcMapQueryLibraryReactor class is used to represent a query library reactor, which notifies an application of query library events.

To create a specific query library reactor, subclass from AcMapQueryLibraryReactor.

Use AcMapQueryLibrary::AddReactor to add a reactor to a query library and AcMapQueryLibrary::RemoveReactor to remove it.

# AcMapQueryCondition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

The AcMapQueryCondition class is derived from the AcMapQueryUnit class. It is the base class from which the following condition classes are derived:

- AcMapSQLCondition
- AcMapLocationCondition
- AcMapDataCondition
- AcMapPropertyCondition

Do not subclass from this class.

# AcMapDataCondition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

A data condition is used to select drawing objects based on object data, which is nongraphic data stored about the drawing in the drawing file.

A data condition contains a [BSPSPopupOnMouseOver\(event\)::">join operator](#), a data type, a [BSPSPopupOnMouseOver\(event\)::">condition operator](#), and a value. In addition, an associated value expression specifies the table and column in which the drawing's [BSPSPopupOnMouseOver\(event\)::">object data](#) is stored, so that the object data can be attached before the query is executed.

Drawing objects whose related object data values meet the conditions of the data condition expression are selected. For example, to select drawing objects that represent pipes of a diameter of one inch, create a data condition that queries a table called PIPES and select those drawing objects whose DIAMETER column value = 1.0.

After you have built the data condition, add it to the query branch using the `AcMapQueryBranch::AppendOperand`, `AcMapQueryBranch::InsertOperand`, or `AcMapQueryBranchSubstituteOperand` function. Since these functions make a copy of the `AcMapDataCondition` object, you can release the data condition after adding it to the query branch.

The `AcMapDataCondition` class is derived from the `AcMapQueryCondition` class.

Do not subclass from this class.

# AcMapLocationCondition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

A location condition is used to select drawing objects based on their spatial location in a drawing.

A location condition contains a join operator (AND or OR), a location type (either Inside or Crossing), and a boundary.

There are many types of boundaries: location boundaries, all boundaries, circle boundaries, fence boundaries, buffer fence boundaries, polygon boundaries, window boundaries, polyline boundaries, buffer polyline boundaries, and closed polyline boundaries.

Inside and Crossing are not appropriate with all types of boundaries. For example, you cannot use Inside with a fence boundary, only Crossing.

After you have created the location condition, add it to a query branch using the `AcMapQueryBranch::AppendOperand`, `AcMapQueryBranch::InsertOperand`, or `AcMapQueryBranchSubstituteOperands`. Since these functions make a copy of the `AcMapLocationCondition` object, you can release the location condition after adding it to the query branch.

After you define the query with the query branch, the query will select drawing objects that lie inside or cross the boundary, depending on the location condition's location type.

The `AcMapLocationCondition` class is derived from the `AcMapQueryCondition` class.

Do not subclass from this class.

# AcMapPropertyCondition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

A property condition is used to select drawing objects based on their properties in the drawing.

A property condition contains a join operator (AND or OR), a [property type](#), a [= 4](#)) [BSPSPopupOnMouseOver\(event\);">comparative operator](#), and a value. Drawing objects whose property values meet the conditions of the property condition expression are selected by the query. For example, in the condition

```
COLOR = 'RED' AND ELEVATION > 50
```

the join operator is AND, the property types are COLOR and ELEVATION, the comparative operators are = and >, and the values are "RED" and 50.

After you create a property condition, add it to a query branch using the `AcMapQueryBranch::AppendOperand`, the `AcMapQueryBranch::InsertOperand`, or `AcMapQueryBranch::SubstituteOperand` function. Since these functions make a copy of the `AcMapPropertyCondition` object, you can release the property condition after adding it to the query branch.

The `AcMapPropertyCondition` class is derived from the `AcMapQueryCondition` class.

Do not subclass from this class.

# AcMapSQLCondition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapQuery.h.

An SQL condition is used to select drawing objects based on nongraphic data that is stored in an external database.

An SQL condition contains a [= 4\) BSPSPopupOnMouseOver\(event\);">join operator](#) and an SQL WHERE condition. In addition, an SQL condition specifies an associated link path name, called a [= 4\) BSPSPopupOnMouseOver\(event\);">link template](#), to the external database in which the data is stored, so that the drawing and its related records can be compared during query execution.

After you have created an SQL condition, add it to a query branch using the `AcMapQueryBranch::AppendOperand`, `AcMapQueryBranch::InsertOperand`, or `AcMapQueryBranch::SubstituteOperand` function. Since these functions make a copy of the `AcMapSQLCondition` object, you can release the SQL condition after adding it to the query branch.

After you define the query with the query branch, the query will select drawing objects whose related database record values meet the conditions of the SQL WHERE condition. For example, to select all drawing objects that represent pressure valves, you might create a SQL condition such as `VALVE_TYPE = 'PRESSURE'`. The WHERE keyword is always omitted from such a statement.

The `AcMapSQLCondition` class is derived from the `AcMapQueryCondition` class.

Do not subclass from this class.

# AcMapAllBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapAllBoundary class is derived from the AcMapLocationBoundary class. It represents the All Boundary condition. Use an object of the AcMapAllBoundary class to select all the objects in the active drawing.

Do not subclass from this class.

# AcMapBufferFenceBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapBufferFenceBoundary class defines a buffer fence, which is a series of points connected by straight lines, surrounded by a buffer with a specified width. The buffer fence boundary is defined by the coordinates of its points and the width of its buffer. There must be a minimum of two distinct points. If the width of the buffer is 0, the buffer fence boundary is treated as a fence boundary.

A query built on a buffer fence boundary can select all drawing objects that cross or lie inside the boundary.

The AcMapBufferFenceBoundary class is derived from the AcMapFenceBoundary class. To get or set buffer fence boundary points, use the AcMapFenceBoundary::GetPoints or AcMapFenceBoundary::SetPoints function.

Do not subclass from this class.

You create an AcMapBufferFenceBoundary pointer using the new operator and delete allocated memory using the delete operator.

# AcMapBufferPolylineBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapPolylineBoundary defines a buffer polyline boundary, which is one or more connected line segments or circular arcs, all surrounded by a buffer with a specified width. If the width of the buffer is 0, the buffer polyline boundary is treated as a polyline boundary.

A query can select drawing objects that either cross or lie inside the buffer polyline boundary.

Do not subclass from this class.

You get an AcMapBufferPolylineBoundary pointer using the new operator and you are responsible for deleting allocated memory using the delete operator.

# AcMapCircleBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapCircleBoundary represents a circle with a center point and a radius.

The AcMapCircleBoundary class is derived from the AcMapPointBoundary class. To get or set the circle's center, use AcMapPointBoundary::GetPoint or AcMapPointBoundary::SetPoint function.

Do not subclass from this class.

You get an AcMapCircleBoundary pointer using the new operator, so you are responsible for deleting allocated memory with the delete operator.

# AcMapClosedPolylineBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapPolylineBoundary defines a closed polyline boundary, which is a series of points connected by straight lines and arcs that define a contiguous area. A query can select drawing objects that either cross or lie inside the closed polyline boundary.

The AcMapClosedPolylineBoundary class is derived from the AcMapPolylineBoundary class.

Objects of the the AcMapClosedPolylineBoundary class are fully enclosed, in contrast to an AcMapPolylineBoundary object, which may or may not be closed.

Do not subclass from this class.

You get an AcMapClosedPolylineBoundary pointer using the new operator and you are responsible for deleting allocated memory using the delete operator.

# AcMapFenceBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapFenceBoundary class defines a fence, which is a series of points connected by straight lines. The fence boundary is defined by the coordinates of its points. There must be a minimum of two distinct points.

A query built on a fence boundary can select all drawing objects that cross the fence boundary.

The AcMapFenceBoundary class is derived from the AcMapLocationBoundary class.

The AcMapBufferFenceBoundary, AcMapPolygonBoundary, and AcMapWindowBoundary classes are derived from the AcMapFenceBoundary class.

Do not subclass from this class.

You get an AcMapFenceBoundary pointer using the new operator, so you are responsible for deleting allocated memory with the delete operator.

# AcMapLocationBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapLocationBoundary class is an abstract base class from which all other location boundary classes are derived. Classes derived from the AcMapLocationBoundary class are used to create instances of different types of boundaries. Boundary types include all boundaries, fence, circle, point, polygon, polyline, and window.

A location condition displays objects based on location relative to a specified boundary, represented by an instance of a location boundary subclass. The query can specify whether objects must be completely inside the boundary or cross and have any part inside the boundary.

To use a location boundary in a query, instantiate an object of one of the AcMapLocationBoundary subclasses, associate it with a location condition using the AcMapLocationCondition::SetBoundary function, and add the location condition to a query branch.

Do not subclass from this class.

# AcMapPointBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapPointBoundary represents the point boundary condition, which is used to define a point and then build a query that selects all the objects that cross the point.

With some location boundaries, you can specify that objects are selected that either cross (AcMap::kLocationCrossing) or lie inside of the boundary (AcMap::kLocationInside). However, with the AcMapPointBoundary class, you can only specify AcMap::kLocationCrossing.

The AcMapPointBoundary class is derived from the AcMapLocationBoundary class.

Do not subclass from this class.

You create an AcMapPointBoundary pointer using the new operator, so you are responsible for deleting allocated memory using the delete operator.

# AcMapPolygonBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapPolygonBoundary defines a series of points connected by straight lines that define a polygon. Objects of the AcMapPolygonBoundary class are always fully enclosed, unlike members of the AcMapPolylineBoundary class, which may or may not be closed.

A query can select drawing objects that either cross or lie inside the polygon boundary.

The AcMapPolygonBoundary class is derived from the AcMapFenceBoundary class. To get a AcMapPolygonBoundary object's points, use AcMapFenceBoundary::GetPoints.

Do not subclass from this class.

You get an AcMapPolygonBoundary pointer using the new operator and you are responsible for deleting allocated memory with the delete operator.

# AcMapPolylineBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapPolylineBoundary defines a series of points connected by straight lines that define a polyline. An AcMapPolylineBoundary object may or may not be fully closed. If it is not closed, a query can select only those drawing objects that cross the polyline. If it is closed, a query can select drawing objects that either cross or lie inside the polyline boundary.

Do not subclass from this class.

You get an AcMapPolylineBoundary pointer using the new operator, and you are responsible for deleting allocated memory using the delete operator.

# AcMapWindowBoundary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapBoundary.h.

The AcMapWindowBoundary represents a rectangle based on two points that specify its left bottom and right top corners. A query can select drawing objects that either cross or lie inside the window boundary.

The AcMapWindowBoundary class is derived from the AcMapPolygonBoundary class, which is derived from the AcMapFenceBoundary class. To get window boundary points, use AcMapFenceBoundary::GetPoints.

Do not subclass from this class.

You create an AcMapWindowBoundary pointer using the new operator and you are responsible for deleting allocated memory using the delete operator. For simplicity, error checking is not shown.

# AcMapPropertyAlteration class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapAlteration.h.

An instance of the `AcMapPropertyAlteration` class represents a property alteration. A property alteration has an alteration type, which identifies the property to be altered (color, layer, line type, and so on), and an expression, which describes how the property is to be altered when the results of a query are displayed with property alteration enabled.

Created by calling `AcMapPropertyAlterationDefinition::AddAlteration`. This function returns a pointer to `AcMapPropertyAlteration` that you are responsible for deleting with the delete operator.

A collection of property alterations constitutes a property alteration definition, which is represented by an `AcMapPropertyAlterationDefinition` object. The property alteration definition is contained by the query to which it applies.

There are two other classes for representing property alterations: `AcMapTextAlteration` and `AcMapHatchAlteration`. Note that these two are subclassed from `AcMapPropertyAlteration`, and that both of them use the `SetExpression` function of the base class to set their respective expressions—to specify the string for the text alteration, or the hatch pattern name for the hatch alteration.

Do not subclass from this class.

The following code alters the color of queried objects to red. This example assumes that a drawing is already attached. For simplicity, error checking is not shown.

```
AcMapSession      *mapApi = NULL;
AcMapProject      *pProj = NULL;
AcMapDrawingSet  *pDSet = NULL;
AcMapQueryBranch *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery       *pQuery = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetDrawingSet(pDSet);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
pQuery->GetPropertyAlteration(pDef);
pDef->AddAlteration(pcAlt, AcMap::kAlterationColor);
pcAlt->SetExpression("RED");
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
pDSet->ZoomExtents();
delete pcAlt ;
delete pLocationCondition;
```

```
delete pQBranch;  
delete pQuery;
```

# AcMapPropertyAlterationDefinition class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapAlteration.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapPropertyAlterationDefinition class represents a property alteration definition, which is an ordered collection of property alterations. These property alterations affect the appearance of objects displayed in the results of a query if property alteration for the query is enabled.

There are three classes for representing property alterations: AcMapPropertyAlteration, AcMapTextAlteration, and AcMapHatchAlteration. The latter two are subclassed from the first, and both of them use the parent's SetExpression function to specify the string for the text alteration or a hatch pattern name for the hatch alteration.

To create or get the property alteration definition of a query, use the AcMapQuery::GetPropertyAlteration function. If property alteration is not yet defined for the query, calling this function creates an empty definition that you can complete by adding property alterations.

Do not subclass from this class. Do not delete the AcMapPropertyAlterationDefinition memory.

The following code alters the color of queried objects to red. This example assumes that a drawing is already attached. For simplicity, error checking is not shown.

```
AcMapSession      *mapApi = NULL;
AcMapProject      *pProj = NULL;
AcMapDrawingSet  *pDSet = NULL;
AcMapQueryBranch *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery *pQuery = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetDrawingSet(pDSet);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
pQuery->GetPropertyAlteration(pDef);
pDef->AddAlteration(pcAlt, AcMap::kAlterationColor);
pcAlt->SetExpression("RED");
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
pDSet->ZoomExtents();
delete pcAlt ;
delete pLocationCondition;
delete pQBranch;
delete pQuery;
```



# AcMapHatchAlteration class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapAlteration.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapHatchAlteration class represents a hatch alteration object, which provides hatch patterns for closed figures when they are queried into the project drawing if property alteration is enabled for the query.

To access members of this class, create an AcMapHatchAlteration pointer using AcMapPropertyAlterationDefinition::AddAlteration and cast it to the AcMapHatchAlteration\* type as follows:

```
pDef->AddAlteration(pcAlt, AcMap::kAlterationHatch);

...
AcMapTextAlteration *pHatch = NULL;
pHatchAlt = (AcMapHatchAlteration*)pcAlt;
pHatchAlt->SetColor("RED");
```

You are responsible for deleting the pcAlt pointer. Do not attempt to delete the pDef pointer.

The AcMapHatchAlteration class is a subclass of the AcMapPropertyAlteration class. Note that it inherits certain functions of the parent class without overloading them. In particular, note that it uses the parent's SetExpression function to specify the hatch pattern name for the hatch alteration.

Do not subclass from this class.

You create an AcMapHatchAlteration pointer by adding an alteration to the property alteration definition. Next, you specify the hatch pattern using AcMapTextAlteration::SetExpression. Default values are used to alter the pattern unless you access members of AcMapHatchAlteration to you change these values, as shown in the following example, which changes the color of the hatch pattern on the WATER layer. This example assumes you have attached citymap7.dwg from the MAPTUT directory.

```
// For simplicity, error checking is not shown.

AcMapProject *pProj = NULL;
AcMapQueryBranch *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery *pQuery = NULL;

AcMapSession *mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
pQuery->GetPropertyAlteration(pDef);
```

```
pDef->AddAlteration(pcAlt, AcMap::kAlterationHatch);
AcMapHatchAlteration *pHatchAlt = NULL;
pHatchAlt = (AcMapHatchAlteration*)pcAlt;
pHatchAlt->SetLayer("WATER");
pHatchAlt->SetColor("CYAN");
pcAlt->SetExpression("Solid") ;
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
delete pcAlt ;
delete pLocationCondition;
delete pQBranch;
delete pQuery;
```

# AcMapTextAlteration class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapAlteration.h.

**Note** This class has [new functions](#) added for AutoCAD Map 3D 2005.

An instance of the AcMapTextAlteration class represents a text alteration, which provides text labels for objects queried into the project drawing if property alteration is enabled for the query.

The AcMapTextAlteration class is a subclass of the AcMapPropertyAlteration class. Do not subclass from this class. Note that it inherits certain functions of the base class without overloading them.

AcMapTextAlteration inherits AcMapTextAlteration::GetType from AcMapPropertyAlteration and it returns kAlterationTextEntity.

To access the members of this class, create an AcMapTextAlteration pointer using AcMapPropertyAlterationDefinition::AddAlteration and cast it to the AcMapTextAlteration\* type as follows:

```
pDef->AddAlteration(pcAlt, AcMap::kAlterationTextEntity);

...
AcMapTextAlteration *pTextAlt = NULL;
pTextAlt = (AcMapTextAlteration*)pcAlt;
pTextAlt->SetColor("RED");
```

You are responsible for deleting the pcAlt pointer. Do not attempt to delete the pDef pointer.

You create an AcMapTextAlteration pointer by adding an alteration to the property alteration definition. Next, you specify the text to be altered using AcMapTextAlteration::SetExpression. Default values are used to alter the specified text unless you access members of AcMapTextAlteration, as shown in the following example that changes the color of text. This example assumes you have attached citymap7.dwg from the MAPTUT directory.

```
// For simplicity, error checking is not shown.

AcMapSession *mapApi = NULL;
AcMapProject *pProj = NULL;
AcMapQueryBranch *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery *pQuery = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
    AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
pQuery->GetPropertyAlteration(pDef);
```

```
pDef->AddAlteration(pcAlt, AcMap::kAlterationTextEntity);
AcMapTextAlteration *pTextAlt = NULL;
pTextAlt = (AcMapTextAlteration*)pcAlt;
pTextAlt->SetColor("RED");
pcAlt->SetExpression(":NAME@WATER_BODIES");
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
delete pcAlt ;
delete pLocationCondition;
delete pQBranch;
delete pQuery;
```

# AcMapRangeLibrary class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

The AcMapRangeLibrary class represents a range library, which contains all the range tables available to a project.

A range library is instantiated automatically when a project is instantiated. There is only one range library per project. You get a pointer to the project's range library with the AcMapProject::GetRangeLibrary function. Do not delete the memory associated with the AcMapRangeLibrary pointer.

Do not subclass from this class.

The following example shows how to use this class. It assumes that you have already attached citymap7.dwg from the MAPTUT directory. It queries all objects in the drawing, and labels each polyline and line. For simplicity, error checking is not shown.

```
AcMapSession      *mapApi = NULL;
AcMapProject      *pProj = NULL;
AcMapDrawingSet   *pDSet = NULL;
AcMapQueryBranch  *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery        *pQuery = NULL;
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
AcMapRangeLibrary *pRangeLib = NULL;
AcMapRangeTable  *pTable = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetDrawingSet(pDSet);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
pQuery->GetPropertyAlteration(pDef);
pDef->AddAlteration(pcAlt, AcMap::kAlterationTextEntity);
pProj->GetRangeLibrary(pRangeLib);
pRangeLib->Clear();
const char *pcName = "MyTypeRangeTable";
const char *pcDsc = "Table for types";
pRangeLib->AddRangeTable(pTable, pcName, pcDsc);
pTable->AddRangeLine(AcMap::kRangeEq, "polyline", "PLINE");
pTable->AddRangeLine(AcMap::kRangeEq, "line", "LINE");
pTable->AddRangeLine(AcMap::kRangeEq, "lwpolyline", "LWPOLYLINE");
pcAlt->SetExpression("(Range .TYPE MyTypeRangeTable)");
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
pDSet->ZoomExtents();
```

```
delete pTable ;  
delete pcAlt ;  
delete pLocationCondition;  
delete pQBranch;  
delete pQuery;
```

# AcMapRangeLine class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

The AcMapRangeLine class represents a range line, also called a range expression. A range line is a line in a range table consisting of a condition to compare to an object property or object data value and the value to return if the condition evaluates to True.

The condition portion of the range line consists of an operator and a simple expression value. The operator (AcMap::ERangeOperator) can be a comparative operator or a special otherwise operator (AcMap::kRangeOtherwise) which covers conditions not specified by the other range lines in the range table. The expression value is called the range definition.

For example, in the range line

```
> 1000 SMALL
```

> is the range operator, 1000 is the range definition, and SMALL is the return value. In the UI, the user would see this range line displayed as

```
If > 100 Return: SMALL
```

Create a range line using the AcMapRangeTable::AddRangeLine function. Get a pointer to AcMapRangeLine using the AcMapRangeTable::GetRangeLine function. You are responsible for deleting the memory allocated for the range line using the delete operator.

To associate range lines in a range table with a property alteration, pass an expression that evaluates to the range table to AcMapPropertyAlteration::SetExpression. For example:

```
//Add a range table to the range library
AcMapRangeTable *pTable = NULL;
const char *pcName = "MyTypeRangeTable";
const char *pcDsc = "Table for types";

if(pRangeLib->AddRangeTable(pTable, pcName, pcDsc) == AcMap::kOk)
{
    // Add range lines that will add text next to the entities
    // that match the input criteria.
    pTable->AddRangeLine(AcMap::kRangeEq, "10e", "LOW");
    pTable->AddRangeLine(AcMap::kRangeEq, "20", "MEDIUM");
    pTable->AddRangeLine(AcMap::kRangeEq, "30", "HIGH");
    pTable->AddRangeLine(AcMap::kRangeOtherwise, NULL,
        "Out of range.");
}

//Add a Text Alteration
AcMapPropertyAlteration *pPropAltObj = NULL;
if(pPADef->AddAlteration(pPropAltObj, AcMap::kAlterationTextEntity) == AcMap::kOk)
{
    AcMapTextAlteration *pTextAlt = NULL;

    //cast the property alteration to a text alteration
    pTextAlt = (AcMapTextAlteration*)pPropAltObj;

    //set some attributes
    pTextAlt->SetColor("Magenta");
}
```

```
pTextAlt->SetJustification("MIDDLE");
pTextAlt->SetRotation("45.0");
pTextAlt->SetHeight("0.5");

// Set the expression and supply the range table,
// "MyTypeRangeTable".
pTextAlt->SetExpression("(Range .TYPE MyTypeRangeTable)");
}
```

Do not subclass from this class.

For more information, click [□](#). Also click Using AutoCAD Map > Queries > Altering the Properties of Queried Objects > Creating a Range Table on the Contents tab of AutoCAD Map Help.

# AcMapRangeTable class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

The AcMapRangeTable class represents a range table, which contains a collection of range lines (also called range expressions).

Get an AcMapRangeTable pointer using the AcMapRangeLibrary::AddRangeTable function, and delete the memory allocated by this function using the delete operator.

Do not subclass from this class.

The following example shows how to use this class. It assumes that you have already attached citymap7.dwg from the MAPTUT directory. It queries all objects in the drawing, and labels each polyline and line. For simplicity, error checking is not shown.

```
AcMapSession      *mapApi = NULL;
AcMapProject      *pProj = NULL;
AcMapDrawingSet   *pDSet = NULL;
AcMapQueryBranch  *pQBranch = NULL;
AcMapLocationCondition *pLocationCondition = NULL;
AcMapQuery        *pQuery = NULL;
AcMapPropertyAlterationDefinition *pDef = NULL;
AcMapPropertyAlteration *pcAlt = NULL;
AcMapRangeLibrary *pRangeLib = NULL;
AcMapRangeTable   *pTable = NULL;

mapApi = AcMapGetSession();
mapApi->GetProject(pProj);
pProj->GetDrawingSet(pDSet);
pQBranch = new AcMapQueryBranch();
pLocationCondition = new AcMapLocationCondition(AcMap::kOperatorAnd,
        AcMap::kLocationInside);
pLocationCondition->SetBoundary((&AcMapAllBoundary;()));
pProj->CreateQuery(pQuery, Adesk::kFalse);
pQuery->Clear(Adesk::kTrue);
pQuery->SetMode(AcMap::kQueryDraw);
pQBranch->AppendOperand(pLocationCondition);
pQuery->GetPropertyAlteration(pDef);
pDef->AddAlteration(pcAlt, AcMap::kAlterationTextEntity);
pProj->GetRangeLibrary(pRangeLib);
pRangeLib->Clear();
const char *pcName = "MyTypeRangeTable";
const char *pcDsc = "Table for types";
pRangeLib->AddRangeTable(pTable, pcName, pcDsc);
pTable->AddRangeLine(AcMap::kRangeEq, "polyline", "PLINE");
pTable->AddRangeLine(AcMap::kRangeEq, "line", "LINE");
pTable->AddRangeLine(AcMap::kRangeEq, "lwpolyline", "LWPOLYLINE");
pcAlt->SetExpression("(Range .TYPE MyTypeRangeTable)");
pQuery->EnablePropertyAlteration(Adesk::kTrue);
pQuery->Define(pQBranch);
pQuery->Run();
pDSet->ZoomExtents();
delete pTable ;
```

```
delete pcAlt ;  
delete pLocationCondition;  
delete pqBranch;  
delete pqQuery;
```

# AcMapErrorEntry class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

An instance of the AcMapErrorEntry class represents an error. Error entries are pushed onto an error stack where you can retrieve them.

You do not create error entries explicitly. AutoCAD Map automatically generates an error entry whenever it encounters an error. You call AcMapErrorStack::GetEntry to get an error entry object pointer.

Do not subclass from this class.

# AcMapErrorParameter class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

An instance of the AcMapErrorParameter class stores information about an error entry parameter. You get an AcMapErrorParameter pointer by calling AcMapErrorEntry::GetParameter, and you must delete the memory allocated for the object using the delete operator.

Do not subclass from this class.

# AcMapErrorStack class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapArxApi.h.

An instance of the AcMapErrorStack class represents the error stack for an AutoCAD Map session.

The error stack is a last-in, first-out data structure, where the most recent error is pushed onto the top of the stack. The AcMapErrorStack::GetEntry function pops the most recent error from the top of the stack.

AutoCAD Map clears the error stack after completing a command.

AutoCAD Map automatically generates an error entry whenever it encounters an error and pushes the error entry onto the error stack. An application can also push error entries onto the error stack.

Do not subclass from this class.

# AcMapObjArray class

[Functions](#) [Object Model](#) [See Also](#)

This template class is defined in MapColl.h.

```
template <class ApiObj> class AcMapObjArray
```

The AcMapObjArray class is used for dynamic object arrays. It stores ApiObj objects in the member data.

To use this template for collecting objects of a particular class, you must define the operator= function for the class.

The AutoCAD Map ARX API uses AcMapObjArray and AcMapObjPtrArray instead of MFC collections to provide array services.

A dynamic object array can grow as needed at runtime.

# AcMapObjPtrArray class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapColl.h.

```
template <class ApiObj> class AcMapObjPtrArray
```

The AcMapObjPtrArray class is used for dynamic object pointer arrays. It stores pointers to ApiObj objects in the member data. The application is responsible for allocating and freeing these objects.

To use this template for collecting objects of a particular class, you must define the operator= function for the class.

The AutoCAD Map ARX API uses AcMapObjPtrArray and AcMapObjArray instead of MFC collections to provide array services.

A dynamic object array can grow as needed at runtime.

# AcMapStringArray class

[Functions](#) [Object Model](#) [See Also](#)

Header file: MapStringArray.h.

The AcMapStringArray class is a dynamic array used to represent character strings. A dynamic string array can grow as needed at runtime.

The AutoCAD Map ARX API uses AcMapStringArray instead of MFC CStringList for dynamic character strings.

# AcMapGetSession Global Function

Gets the AutoCAD Map application object.

Include MapSession.h.

```
AcMapSession*  
AcMapGetSession();
```

Returns the application object.

Before you can do anything with AutoCAD Map, you must get the application object.

```
AcMapSession* pSession = AcMapGetSession();
```

## AcMap::EAdeDwgStatus

Attached drawing status codes.

0	kDwgInactive
1	kDwgActive
4	kDwgLocked

## AcMap::EAdeDwgUpdateStatus

Drawing update status.

0	kDwgNonUpdated	Nothing to save.
1	kDwgShouldBeReloaded	Lock information updated. Another ADE should only reload the drawing.
4	kDwgShouldBeRequeried	Queried objects have been modified. Another ADE instance should requery.

## AcMap::EAlterationType

Property alteration types.

00	kAlterationBlockName	Block name.
01	kAlterationColor	Color.
02	kAlterationLayer	Layer name.
03	kAlterationRotation	Rotation.
04	kAlterationElevation	Z coordinate in the user coordinate system.
05	kAlterationHeight	Text height.
06	kAlterationLineType	Line type.
07	kAlterationScale	Scaling factor. For example "1.2" = 120%
08	kAlterationStyle	Text style.
09	kAlterationText	Text value.
10	kAlterationThickness	Thickness.
11	kAlterationWidth	Line width.
12	kAlterationTextEntity	Text entity definition.
13	kAlterationHatch	Hatch definition.

# AcMap::EClassId

Run-time class identification for query condition classes

## Location boundaries

01	kLocationBoundary
02	kAllBoundary
03	kPointBoundary
04	kCircleBoundary
05	kFenceBoundary
06	kBufferFenceBoundary
07	kPolygonBoundary
08	kWindowBoundary
09	kPolylineBoundary
10	kBufferPolylineBoundary
11	kClosedPolylineBoundary

## Query units

12	kQueryUnit	Query unit of undetermined type.
13	kQueryCondition	Query condition of undetermined type.
14	kLocationCondition	Location condition.
15	kPropertyCondition	Property condition.
16	kSQLCondition	SQL condition.
17	kDataCondition	Data condition.
18	kLocationConditionImp	Not intended for public use.
19	kPropertyConditionImp	Not intended for public use.

20	kSQLConditionImp	Not intended for public use.
21	kDataConditionImp	Not intended for public use.
22	kQueryBranch	Query branch, a set of grouped conditions.

## AcMap::EConditionOperator

Comparison operators for query conditions.

**Note** The only valid operator in a string context is `AcMap::kCondEq`.

1	<code>kCondEq</code>	Equal.
2	<code>kCondGT</code>	Greater than.
3	<code>kCondGTorEq</code>	Greater than or equal.
4	<code>kCondLT</code>	Less than.
5	<code>kCondLTorEq</code>	Less than or equal.
6	<code>kCondNotEq</code>	Not equal.

## AcMap::EDataQueryType

Data query types.

0	kDataIRD	Internal resources dictionary data tables, object data fields, and an expression
1	kDataAttribute	Blocks and attribute tags
2	kDataLinkTemplate	Link templates and key columns for tables
3	kDataEED	Extended entity data
4	kDataFeature	Feature data

# AcMap::EDataType

Data types for object data fields.

0	kUnknownType
1	kInteger
2	kReal
3	kCharacter
4	kPoint

# AcMap::EErrCode

[See Also](#)

AcMap::EErrCode enumerators represent error codes.

Errors on the error stack are represented by AcMapErrorEntry objects. Error codes are returned by the AcMapErrorEntry::ErrorCode function.

[-15](#)    [1200](#)    [2000](#)    [2500](#)    [3000](#)  
[0](#)    [1300](#)    [2100](#)    [2600](#)    [3100](#)  
[200](#)    [1500](#)    [2200](#)    [2700](#)    [3200](#)  
[1000](#)    [1800](#)    [2300](#)    [2800](#)    [4000](#)  
[1100](#)    [1900](#)    [2400](#)    [2900](#)

## Expression

-15	kErrXEDValueFail	Internal error.
-14	kErrRngTabEvalFail	Internal error.
-13	kErrNoRngTabFound	Internal error.
-12	kErrNoRngTabLibFound	Internal error.
-11	kErrLispValueFail	Internal error.
-10	kErrGetAttrFail	Internal error.
-09	kErrGetSQLFail	Internal error.
-08	kErrGetEEDFail	Internal error.
-07	kErrInvalidProperty	Internal error.
-06	kErrExpInvalidOperand	Internal error.
-05	kErrExpADS	Internal error.
-04	kErrExpNoMemforOperand	Internal error.
-03	kErrExpMathOverFlow	Internal error.
-02	kErrExpInvalidOper	Internal error.
-01	kErrExpSyntaxErr	Internal error.

---

---

For more Expression errors, which begin at 2900, click [□](#).

## Common Usage

00	kOk	General ADE return value: call to ADE object is not successful.
01	kErr	Invalid function argument.
02	kErrBadInput	
03	kErrObjectNotFound	
04	kErrOutOfMemory	
05	kErrObjNotInitialized	
06	kErrWrongType	
07	kErrWrongProject	
08	kErrEOB	Invalid function argument type.
09	kErrADSError	
10	kErrAdsNameConversionFails	
11	kErrWrongArgument	
12	kErrWriteBufFails	
13	kErrReadBufFails	
14	kErrXDataCorrupted	
15	kErrNoEnvironment	
16	kErrUsrBreak	
17	kErrUncomparable	
18	kErrPermissionDenied	No permission to perform some action. When using the following commands, the message has different meanings: ADESAVEOBS = can't be executed in demonstration mode; ADEDRAWINGS = User has no rights to update the set; ADESELOBS, ADDREMOBS = User has no rights to edit; ADEDWGMAINT = User needs to be superuser to remove the foreign locks; MAPOPTIONS: System Preferences = can't set "Force User Login" flag

and change "Object Locking" flag if drawing set contains active or locked drawings. Deactivate or unlock the drawings before trying to set these preferences.

19	kErrWrongSyntax	
20	kErrDuplicate	
21	kErrInvalidPathOrFileName	
22	kErrInvalidVersion	
23	kErrFileIOFatalError	

## External Subsystems

200	kErrAcDbError	
201	kErrIRDNNotInitialized	
202	kErrIRDError	
203	kErrASENotInitialized	
204	kErrASEError	
205	kErrASIErr	
206	kErrAsiNotInitialized	
207	kErrAsiConnectToEnvironmentFailed	

## Transactional Manager

1000	kErrClosed	Repeated attempts to close previously closed ADE object. Call support.
1001	kErrWasErased	Attempt to work with erased ADE object. Call support.
1002	kErrOpenForRead	Attempt to update ADE objects opened for read. Call support.
1003	kErrOpenForWrite	Attempts to get multiple access to ADE object opened for write. Call support.
1004	kErrWrongMode	Incorrect mode of the ADE objects should be opened. Call support.
1005	kErrClone	Exception at the time of cloning ADE objects. Call support.

1006	kErrResponse	Incorrect attempt to start ADE transaction. Call support.
1007	kErrObjIsAbsent	Attempt to work with erased ADE object. Call support.
1008	kErrAccess	Type of work with ADE does not correspond to its status. Call support.
1009	kErrMultipleUsage	Attempts to get multiple access to ADE object opened for write. Call support.
1010	kErrUpgrade	Incorrect attempt to update opening mode of the ADE object. Call support.
1011	kErrNotClosed	Object was not closed during current ADE transaction. Call support.

### CAdeList

1100	kErrGetIteratorFails
1101	kErrListIsEmpty

### CAdeListIterator

1102	kErrListEnd
1103	kErrListObjectIsAbsent

### Drawing

1200	kErrQueriedEntity	Ignoring queried entity for saving selection. When using ADESAVEOBS, the message means queried objects that are selected for save back are ignored.
1201	kErrDwgNotActive	Attempt to read the object locked from an inactive drawing. When using ADEWHOASIT, the message means the drawing from which the object was queried is no longer active. ADE is unable to determine if the object selected is currently locked. Activate the specified drawing and reenter the command.
1202	kErrReadDwgFileFails	ADE fails to read external drawing. If using ADEDRAWINGS, this message means that ADE is unable to read the specified drawing. Perhaps the

		drawing doesn't exit or the specified file is not a valid AutoCAD Drawing File. Check the error message stack for more information.
1203	kErrMultipleUsers	ADE fails to access external drawing.. If using ADEDRAWINGS, this message means ADE is unable to access the specified drawing because of file locking problems. Check the error message stack for more information.
1204	kErrEntityLockingFails	ADE fails to lock. If using ADESELOBJECTS, this message means ADE is unable to lock an object. The object may already be locked. Use the ADEWHOHASIT to determine if the object is locked, perhaps by another user. Check the error message stack for more information.
1205	kErrUnlockedEntity	Entity is unlocked by another ADE user. If using ADEWHOHASIT, this message indicates that the selected object is not currently locked.
1206	kErrLockedEntity	Entity has been locked by another ADE user. If using ADESELOBJECTS, this message indicates that the object is already locked by another user. Use the ADEWHOHASIT command to identify the user.
1207	kErrAlreadyInSaveSet	
1208	kErrAlreadyNotInSaveSet	
1209	kErrMultipleReaders	
1210	kErrOpenPrefDictionaryFails	ADE fails to open ADE preferences dictionary. Call support.
1211	kErrSavePrefDictionaryFails	ADE fails to save ADE preferences dictionary. Call support.
1212	kErrRestoreDSetFails	ADE fails to restore the drawing set. Call support.
1213	kErrOpenDSetDictionaryFails	ADE fails to open ADE drawing set dictionary. Call support.
1214	kErrSaveDSetDictionaryFails	ADE fails to save ADE drawing set in dictionary. Call support.
1215	kErrOpenQueryDictionaryFails	ADE fails to open ADE query library dictionary. Call support.

1216	kErrSaveQueryDictionaryFails	ADE fails to save ADE query library in dictionary. Call support.
1217	kErrOpenRTableDictionaryFails	ADE fails to open ADE range table dictionary. Call support.
1218	kErrSaveRTableDictionaryFails	ADE fails to save ADE range table in dictionary. Call support.
1219	kErrRestoreRTableDictionaryFails	ADE fails to restore ADE range table in dictionary. Call support.
1220	kErrOpenDocViewDictionaryFails	ADE fails to open ADE Doc View information dictionary. Call support.
1221	kErrSaveDocViewDictionaryFails	ADE fails to save ADE Doc View information in dictionary. Call support.
1222	kErrOpenKeyViewDictionaryFails	ADE fails to open ADE Key View information dictionary.
1223	kErrSaveKeyViewDictionaryFails	ADE fails to save ADE Key View information in dictionary. Call support.
1224	kErrSaveProjectionFails	ADE fails to save projection code in the drawing. Call support.
1225	kErrCopyHardPointerFails	ADE fails to apply property alteration for some symbol table. Call support.
1226	kErrDwgToBeReloaded	
1227	kErrDwgHasBeenModified	There were objects queried from the drawing that will be treated as new objects. If using ADEDRAWINGS, this message means that when a drawing from which objects have been queried is detached, ADE converts the objects into newly created objects. When you use the ADEWHOHASIT command to see the origin of these objects, it says that they have not been queried.
1228	kErrOnLockedLayer	Objects from a locked layer have been selected. If using ADESELOBJECTS, ADEREMOJECTS, or ADESAVEOBJECTS, this message means you selected objects from a locked layer and they cannot be added to, saved to, or removed from the save set or saved to source.
1229	kErrDwgSaveFails	ADE fails to save the source drawing. If using

		ADESAVEOBS, ADESELOBS, ADEREMOBS, OPEN, NEW, or QUIT, or if you're configuring options, or modifying objects, you may get this message. Call support. NOTE: Use this message exactly as spelled here.
1230	kErrDwgLocksLeft	ADE fails to remove object locks (if present) by the end of ADE session. If using OPEN, NEW, or QUIT, or if you're configuring options, this message means the drawing might be locked by another ADE user.
1231	kErrLinkWillBeLost	ADE does not save links between queried objects and source drawings between ADE sessions. Detach source drawing with queried objects; Use the SAVE command with queried objects or no objects in the save set.
1232	kErrDwgDiskFull	
1233	kErrDwgHasQueriedObject	
1234	kErrUnlockedFile	

### Current Session

1300	kErrGetCPointFails	ADE fails to get Text location for the entity. Call support.
1301	kErrSetCPointFails	ADE fails to store Text location for the entity. Call support.
1302	kErrTextInsPointMissed	Missing Text insert point. Call support.
1303	kErrTextAllignPointMissed	Missing alignment point for Aligned and Fit Text. Call support.
1304	kErrTextHeightMissed	Missing Text height. Call support.
1305	kErrTextStringMissed	Missing Text string. Call support.
1306	kErrIndexUpdateFails	Exception in Index generation or regeneration. If using ADESAVEOBS, ADESELOBS, ADEREMOBS, or ADEDWGMAINT, this message means entity modification occurred. Call support.
1307	kErrIgnorePreview	ADE custom object is selected to add to save set or to save. If using ADESAVEOBS, ADESELOBS, or

		ADEREMOBS this message has the following meaning: When doing a Preview Query, ADE creates a special object called a PREVIEW object, used to display the queried objects. PREVIEW cannot be saved back to a source drawing. When a user selects this object for adding to the save set or when saving back, ADE detects this and prevents the operation.
1308	kErrIgnorePSPACE	
1309	kErrEraseIRDObjectFails	ADE fails to erase the Object Data Table. Call support.
1310	kErrRenameIRDObjectFails	ADE fails to rename the Object Data Table. Call support. If using ADEDEFDATA, this message means ADE can't rename Object Data Table. Perhaps Object Data of the same name already exists in the drawing or Object Data Table definitions bearing the old name are different in the source drawings. Check the error message for more information.
1311	kErrAlterIRDObjectFails	ADE fails to alter Object Data Table. If using ADEDEFDATA, this message means ADE can't alter Object Data Table. Perhaps Object Data Table definitions are different in the source drawings. Check the error message stack for more information.
1312	kErrIrdDuplicateTableName	Object Data Table with specified name already exists. ADEDEFDATA - all Object Data table names must be unique. Duplicate names are not allowed.
1313	kErrIrdNotIdenticalFormat	Conflict in Object Data Table definition. If using ADEDEFDATA, this message means that perhaps Object Data Table definitions are different in the source drawings. NOTE: Use this message exactly as spelled here.
1314	kErrBHatchUnit	ADE treats bhatch and its boundary as one unit. If using ADESELOBS or ADEREMOBS, this message has the following meaning: When adding an object to the save set, ADE checks if this object is part of a hatch boundary. If it is, all other objects that form the boundary are added to or removed from the save set. When saving back bhatched areas, ADE always treats the boundary as one object.
1315	kErrReQuery	ADE queried one or more objects twice. ADE does not support UNDO for this operation. If using ADEQUERY or ADERUNXQUERY this message has the following meaning: If a drawing file has been

		modified by another ADE user and if a queried object matches another query, ADE removes the old copy and queries a new copy. This operation can't be undone.
1316	kErrCantAccessFont	
1317	kErrCantAccessImageFile	
1318	kErrUnableRedefineXrefBlock	

## Drawing Set

1500	kErrRestoreDrawingSetFails	ADE fails to read Drawing Set from DWG file. You encounter this message during ADE initialization or when using Open drawing file or ADEDRAWINGS. If using ADEDRAWINGS - attach drawing, the message means "activate on attach" is ON or ade_dsattach(), ade_dswcattach() drawing is corrupted or old ADE version is in use.
1501	kErrDuplicatedDrawing	Attempt to attach the same drawing twice. If using ADEDRAWINGS, this error appears when a drawing with the same name has already been attached to the work session.
1502	kErrNestedDrawing	Prohibited attempt to edit properties of the drawing in the nested drawing set. If using ADEDRAWINGS, this message means a user is not allowed to modify the transformation and save back extents of nested drawings in the work session. These properties can only be modified for top level drawings.
1503	kErrActivateDrawingFails	ADE fails to activate source drawing. ADE was unable to activate a drawing Perhaps the drawing does not exist or it is locked by another user, or the current user doesn't have permission to read the specified drawing file. Check the error message stack for more information.
1504	kErrDeactivateDrawingFails	ADE fails to deactivate source drawing. ADE is unable to deactivate a drawing. Perhaps the drawing is locked by another user, the drawing no longer exists, or there are locked objects in the drawing. Check the error message stack for more information.
1505	kErrLongDrawingDescription	Specified drawing description exceeds 133 symbols. Shorten description.

1506	kErrEntityHasBeenLocked	Drawing with locked entities can't be deactivated. ADEDRAWINGS - the specified drawing cannot be deactivated because it contains locked objects. Remove locks and deactivate.
1507	kErrPreviewNotSupported	ADE 2.0 ignoresTolerance, Body,Ellipse, 3dSolid, Region and Mline. If using ADEQUERY (Preview), ADEQVIEWDWGS, or ADEKEYVIEW, this message means ADE does not support Preview Query of Tolerance, Body,Ellipse, 3dSolid, Region and Mline.
1508	kErrAliasIsInUse	A drive alias of the same name already exists.
1509	kErrActivateDrawingCancelled	AcMapDrawingSetReactor::DrawingToBeActivated returns Adesk::kFalse, the attach operation is cancelled, and the error is returned to the application to handle.
1510	kErrAttachDrawingCancelled	AcMapDrawingSetReactor::DrawingToBeAttached returns Adesk::kFalse, the attach operation is cancelled, and the error is returned to the application to handle.

## Feature Alteration

1800	kErrInvalidFeatureType	Attempt to set an invalid property alteration type. If calling API functions ade_altpsetprop or ade_altpdefine, check for a mistake in property type.
1801	kErrNoListId	Property alteration internal list is invalid. Call support.
1802	kErrInvalidExpType	Attempt made to set an invalid property alteration expression type. Call support.
1803	kErrNoExpression	Property alteration internal object is invalid. Call support.
1804	kErrTextCreationFailed	Property alteration was unable to create a new text object. This message occurs when using queries that alter properties. Check expressions in the text property alteration definition.
1805	kErrHatchCreationFailed	Property alteration was unable to create a new hatch object. This message occurs when using queries that alter properties. Check expressions in the hatch property alteration definition.
1806	kErrInvalidColor	Invalid color passed to property alteration. This message occurs when using queries that alter properties. Check expressions evaluates to a valid AutoCAD color.

1807	kErrInvalidLayer	Invalid layer name. This message occurs when using queries that alter properties. Check expressions evaluates to a valid AutoCAD layer.
1808	kErrInvalidStyle	Invalid style name. This message occurs when using queries that alter properties. Check expressions evaluates to a valid style.
1809	kErrInvalidJustification	The expression for justification in a text property alteration did not evaluate to a valid justification. This message occurs when using queries that alter properties. Check expressions evaluates to a valid AutoCAD justification.
1810	kErrInvalidScale	
1811	kErrNoRangeId	ADE internal object is invalid. Call support.

## Mapping

1900	kErrMapCoincPoint	Coincident points. If using ADERSHEET or ADETRANSFORM this message means either old or new points are coincident. They must be different.
1901	kErrMapWrongScale	Invalid scale. Call support.
1902	kErrMapTransform	Can't transform entity. ADERSHEET, ADETRANSFORM, ADEQUERY, ADESAVEOBS An error appeared at the time of entity transformation. It is high-level error. There must be another error in the stack with more specific information.
1903	kErrMapWrongExtents	Invalid entity extents. If you're using ADETEXTLOC, ADERSHEET, ADEQUERY, ADESAVEOBS, or calling AcDbEntity::getGeomExtents() function you may get this message.
1904	kErrMapWrongPoints	Invalid points number. If you're using ADERSHEET, this message means that the numbers of old and new points are different, or less than 2. Dialog doesn't allow this. C.
1905	kErrMapWrongSelSet	Invalid selection set. Call support.
1906	kErrMapWrongEntityName	Invalid entity name. The entity is open. For

		example, it has been received from the API.
1907	kErrMapOpenEntity	Can't open entity. The entity is open. For example, another application opened the entity.
1908	kErrMapUpgradeEntity	Can't upgrade open. Entity modification occurred.
1909	kErrMapMoveStretchPoints	Can't modify stretch points. Call support.
1910	kErrMapEntityPoint	Can't modify entity points. Call support.
1911	kErrMapCmdecho	Can't change CMDECHO variable. Using ADEFILLPOLYG may produce this message.
1912	kErrMapCecolor	Can't change CECOLOR variable. Using ADEFILLPOLYG may produce this message.
1913	kErrMapHatch	Error in hatch command. Using ADEFILLPOLYG may produce this message.
1914	kErrMapWrongIntersectForPoints	Can't find intersection. Using ADEQUERY or calling AcDbEntity::IntersectWith() function may produce this message.
1915	kErrMapWrongHandle	Wrong entity handle. Call support.
1916	kErrMapNotPolyline	Entity isn't polyline. Call support.
1917	kErrMapIterator	Can't create iterator. Call support.
1918	kErrMapWriteXData	Can't write Xdata. Ensure that Xdata size is 16 KB or less.
1919	kErrMapBuffer	Can't create buffer. If you're using ADEQUERY, to make a location query using a bufference, you may get this message.
1920	kErrMapStretchPoints	Can't get stretch points. If you're using ADETRANSFORM or ADEQUERY or calling AcDbEntity::getStretchPoints() you may get this message.

## Topology

2000	kErrTopInvalidName	Invalid topology name. Occurs during topology creation.
2001	kErrTopExist	Topology already exists. Occurs during topology creation.

2002	kErrTopBuildNet	Error building network topology. Occurs during topology creation.
2003	kErrTopBuildPolygon	Error building polygon topology. Occurs during topology creation.
2004	kErrTopBuildNode	Can't create node. Occurs during topology creation.
2005	kErrTopBuildArc	Can't create link. Occurs during topology creation.
2006	kErrTopBuildCntr	Can't create centroid. Occurs during topology creation.
2007	kErrTopAPIReg	Can't register topology API. Occurs during ADE loading.
2008	kErrTopFuncNotAvail	Function isn't available. Occurs if you're using topology functions of the API.
2009	kErrTopWriteData	Error writing Xdata. Occurs during topology creation and modification.
2010	kErrTopNotExist	Topology doesn't exist. Occurs if you're using topology functions of the API.
2011	kErrTopOverlayType	Wrong overlay type.
2012	kErrTopMakeLayer	Can't create new layer. Using ADEDWGCLEAN produces this message.
2013	kErrTopBlockNotExist	Block doesn't exist.
2014	kErrTopNotOpenForWrite	Topology isn't open for write. Occurs when editing topology.
2015	kErrTopOpenIrdTable	Can't open object data table. Occurs when loading and editing topology.
2016	kErrTopWrongIrdAttr	Invalid object data table. Occurs when loading and editing topology.
2017	kErrTopLoaded	Topology is already loaded. Occurs when loading topology.
2018	kErrTopIncompleteElem	Incomplete topological element. Occurs when editing topology.
2019	kErrTopInvalidColor	Invalid color number. Occurs when using ADEDWGCLEAN and creating topology.

2020	kErrTopInvalidFlag	Invalid flag. Occurs when using ADEDWGCLEAN.
2021	kErrTopInvalidTolerance	Invalid tolerance. Occurs when using ADEDWGCLEAN and creating topology.
2022	kErrTopInvalidCorridor	Invalid corridor width. Occurs when using ADEDWGCLEAN.
2023	kErrTopInvalidOffset	Invalid offset. Occurs when using buffering.
2024	kErrTopInvalidHeight	Invalid marker height. Occurs when using ADEDWGCLEAN.
2025	kErrTopInvalidMarkerType	Invalid marker type. Occurs when using ADEDWGCLEAN.
2026	kErrTopInvalidEntityType	Invalid type for new entities. Occurs when using ADEDWGCLEAN.
2027	kErrTopInvalidErrorType	Invalid error type. Occurs when using ADEDWGCLEAN.
2028	kErrTopIntersection	Intersections detected. Occurs when creating and editing polygon topology.
2029	kErrTopOverlayItself	Can't overlay topology with itself.
2030	kErrTopSourceDwgAccess	Can't access source drawing.
2031	kErrTopSourceDwgNotActive	Source drawing isn't active.
2032	kErrTopSourceDatabaseAccess	Can't access source drawing database.
2033	kErrTopSourceObjectId	Can't get object ID by handle in source drawing database.
2034	kErrTopNotLoaded	Topology isn't loaded.
2035	kErrTopImplicitNode	Node object doesn't exist in node topology.
2036	kErrTopMisplacedNode	Wrong node coordinates.
2037	kErrTopUnreferencedNode	Node isn't referenced in links.
2038	kErrTopUnexistentNode	Link references nonexistent node.
2039	kErrTopMismatchStartNode	Link has invalid ID at the start node.
2040	kErrTopMismatchEndNode	Link has invalid ID at the end node.
2041	kErrTopMisplacedCentroid	Wrong centroid coordinates.

2042	kErrTopMismatchLeftPoly	Link has invalid ID for the left polygon.
2043	kErrTopMismatchRightPoly	Link has invalid ID for the right polygon.
2044	kErrTopUnexistentCentroid	Centroid isn't inside polygon.
2045	kErrTopMultiplyCentroid	Polygon has several centroids inside.
2046	kErrTopWrongPolyQty	Some polygons are incorrect.
2047	kErrTopMismatchPolyArea	Incorrect polygon area.
2048	kErrTopMismatchPolyPerimeter	Incorrect polygon perimeter.
2049	kErrTopOpenSourceDwgTopo	Topology loaded from source drawings can't be open for write.
2050	kErrTopOpenTempTopo	Temporary topology can't be open for write.
2051	kErrTopIdNotExist	Current drawing doesn't have OD table with information about last ID.
2052	kErrTopEmpty	Can't create or load empty topology.
2053	kErrTopWasModified	Topological objects were modified by AutoCAD commands.
2054	kErrTopMultiple	Object belongs to multiple topologies and can't be erased.
2055	kErrTopCalculateOffset	Can't calculate offset. Use default. Occurs when using buffering.
2056	kErrTopZeroOffset	Zero offset. Can't build buffer.
2057	kErrTopDifferentOffset	Offset has different sign for some objects. Can't build buffer.
2058	kErrTopInvalidSelSet	Invalid selection set. Occurs when using the API.
2059	kErrTopCleanNotInit	Cleanup model isn't initialized. Occurs when using the API.
2060	kErrTopCleanNoGroup	There is no current group. Occurs when using the API.
2061	kErrTopCleanInvalidIndex	Invalid error index. Occurs when using the API.
2062	kErrTopCleanNoError	Current error isn't set. Occurs when using the API.
2063	kErrTopTraceLinkNotExist	Link doesn't exist in tracing model. Occurs when

		using the API.
2064	kErrTopTraceNodeNotExist	Node doesn't exist in tracing model. Occurs when using the API.
2065	kErrTopTraceNoPath	Result path isn't calculated. Occurs when using the API.
2066	kErrTopTraceInvalidIndex	Invalid element index. Occurs when using the API.
2067	kErrTopInvalidExpression	Can't process ADE expression. Occurs when using overlay, buffer, dissolve, or tracing command.
2068	kErrTopLockedTable	Can't write into topology OD table. Occurs when using dissolve command.
2069	kErrTopCreateTable	Can't create OD table. Occurs when using topology creation, overlay, buffer, or dissolve commands.
2070	kErrTopCreateTableColumn	Can't add column to OD table. Occurs when using topology creation, overlay, buffer, or dissolve commands.
2071	kErrTopTraceNodesEqual	Start and end nodes are the same. Occurs doing shortest path tracing.
2072	kErrTopTracePathNotExist	Empty path. Occurs during shortest path tracing.
2073	kErrTopTraceFloodNotExist	Empty path. Occurs when tracing floods.
2074	kErrTopRenameDisabled	Can't rename topology, because current drawing has queried objects with OD.
2075	kErrTopDeleteDisabled	Can't delete topology, because current drawing has queried objects with OD.
2076	kErrTopInvalidExtents	

## Topology API

2100	kErrTopApiErrWrongInput	Missing or invalid parameter.
2101	kErrTopApiWrongId	Invalid ID.

## Tracing

2150	kErrTopSprErr	
------	---------------	--

## Query Definition

2200	kErrUnexpectedBuffChar	Invalid character encountered while reading the query definition from the drawing. Options are a) Recover the drawing b) Define and save a new query definition in the drawing.
2201	kErrInvalidIndex	An invalid line number was specified for either grouping or ungrouping of query lines. Specify the correct line number for grouping or ungrouping the lines.
2202	kErrInvalidQueryLine	One or more query lines have been incorrectly defined. May occur when you incorrectly place a parenthesis or an operator in a query line.
2203	kErrInvalidName	Either a query or a query category name is invalid. Ensure that the query or category name conform to the AutoCad symbol name specifications.
2204	kErrEntryAlreadyExists	Either the query or the query category name already exists in the query library. Ensure that the query name is unique within the query library.
2205	kErrEntryInOtherCategory	The query name specified already exists in another category in the query library. Ensure that the query name is unique within the query library.
2206	kErrEntryAndFileAlreadyExist	The file name specified for saving the external query already exists. Choose a different file name.
2207	kErrASISConnectFailed	The connection to the ASI environment required for the SQL query was not made. Use ASE to connect to the environment before attempting to perform the SQL query.
2208	kErrASISstmtPrepareFailed	The call to CAsiExecStm::Prepare failed. correct the table name or the SQL statement specified.
2209	kErrASISrAllocFailed	The call to CAsiCsr::Allocate failed. Look at the ASI error displayed.
2210	kErrASISrOpenFailed	The call to CAsiCsr::Open failed. Look at the ASI error displayed.
2211	kErrInvalidDOName	An invalid Environment, Schema or Catalog name was specified. Set the correct Environment, Catalog, and Schema names.

2212	kErrLPInitFailed	The call to CAseLinkPath::init failed. Look at the ASE error displayed.
2213	kErrColNotFound	Used for the SQL Order-by dialog now obsolete. Call support.
2214	kErrQDefNotInTM	The CAdeQueryDef object was not appended to the Transaction Manager. Internal error.
2215	kErrQryDefnExists	A query definition already exists and a new one cannot be loaded. Clear the existing query definition before loading a new one.
2216	kErrInvalidOperator	Invalid operator defined in query definition. The specified comparison operator is incompatible with operand types. Do not use > with the point type. Check the query definition and change either the operator or operand type.
2217	kErrInvalidPtrnOperator	Invalid operator defined in query definition for pattern value. If value operand is defined as pattern, only "=" comparison operator can be used. Check the query definition and change either operator or operand value.
2218	kErrInvalidField	Non-existent object data field specified. This error occurs when the user specifies the wrong object data field name for a table (if there is no such field in the specified table) in the query definition and executes the query. Check query definition and tables and correct the mistake.
2219	kErrInvalidNotBranch	
2220	kErrInvalidBranch	
2221	kErrUndefinedValue	
2222	kErrInvalidLocationType	
2223	kErrCantLoadExternQuery	

## Query Manager

2300	kErrIntersectFailed	A call to CAseLinkSel::intersectPartialKey failed. Look at the ASE error displayed.
2301	kErrNoTemplate	The query type was specified as report but no report options were defined. Define report options.

2302	kErrASISoreValueFailed	A call to CAsiData ::storeValue failed. Look at the ASI error displayed.
2303	kErrASIGetValueFailed	A call to CAsiData ::getValue failed. Look at the ASI error displayed.

## Utility

2400	kErrLicFatal	Fatal error in ADE license. Call support.
2401	kErrFileNotFound	Can't find associated document. Occurs when using ADEDOCVIEW.
2402	kErrPathNotFound	Can't find executable file. Occurs when using ADEDOCVIEW.
2403	kErrBadFormat	Syntax error in the command line. Occurs when using ADEDOCVIEW.
2404	kErrConvtErr	Error converting ADE 1.0 data to ADE 2.0 data. Occurs when using ADECONVERT.

## Data Dialogs

2450	kErrIRDMismatch
2451	kErrIRDInvalidName
2452	kErrIRDTableExists
2453	kErrInvalidTableName
2454	kErrInvalidAttrName
2455	kErrTopoName
2456	kErrQueriedAndNotNew
2457	kErrNotAdministrator

## GenLink

2500	kErrTagNotFound
2501	kErrTagValueAbsent
2502	kErrIllegalFormat

2503	kErrColMoreThanOne
------	--------------------

## Environment

2600	kErrInitEnv	An error occurred during the initialization of ADE. The cause of this error may be due to errors in loading/initializing ADE user preferences, system preferences, log file, or user list.
2601	kErrCantFindAdeExePath	
2602	kErrINIWrite	
2603	kErrInvalidUserName	The user name specified does not exist in the user list. Use a user name that already exists in the user list or define a new one using User Administration.
2604	kErrLoadUserList	
2605	kErrSaveUserList	
2606	kErrInvalidPswd	The password specified does not match the one specified in the user list for this user. Use the correct password.

## Rx

2700	kErrRxAseLoad	ASE isn't loaded Can't initialize ASE API.
2701	kErrRxAseInit	Object Data module isn't loaded.
2702	kErrRxIrdLoad	Can't initialize Object Data API.
2703	kErrRxIrdInit	Specified Coord system category not found in the library. Call support.

## Projection

2800	kErrNoProjCatFound	Specified Coord system category not found in the library.Call support.
2801	kErrNoDatumFound	Specified Coord system datum not found. Call support.
2802	kErrNoElipFound	Ellipsoid not found in the ellipsoid list. Call support.
2803	kErrNoCoordFound	Specified Coord Sys not found. Call support.

2804	kErrFaileOpenDatumFile	Can't open projection .mp3 file. Occurs when loading ADE.
2805	kErrFaileOpenElipFile	Can't open ellipse file. Occurs when loading ADE.
2806	kErrNoneCoord	Internal code to set “None” projection to the drawing. Call support.

## Expression

2900	kErrNoExpressionFound	Empty expression is specified. Call support.
2901	kErrGetPropFail	This is an internal code to show that entity has no specified property. Call support.
2902	kErrExpEvalFail	ADE fails to evaluate expression. Occurs when executing query with property alteration and executing a property query.
2903	kErrExpMissingQuote	Quotes mismatched in SQL expression. Occurs when executing a SQL query and a query with SQL property alteration. Also occurs when using ADECONVERT.
2904	kErrExpMissingCParen	Parenthesis mismatched. Occurs when executing a query with feature alteration and executing a property query.
2905	kErrExpExceedThreeOper	More than three operands are specified. Occurs when executing a query with property alteration and executing a property query.
2906	kErrRngTabNameExist	Range table with specified name already exists. Call support.
2907	kErrLpnInvalid	
2908	kErrLpnNotFound	
2909	kErrRangeInvalidElse	

For more Expression errors, which begin at –15, click .

## Index

3000	kErrInvalidIndexVersion	The version of the index in the drawing is invalid. The options are: 1)Regenerate the index using drawing maintenance 2)Remove the index using the index removal utility and then re-generate the index.
------	-------------------------	--

3001	kErrIndexOutOfDate	The index in the drawing is out-of- date. Regenerate the index using drawing maintenance.
3002	kErrTypeAllObjects	
3003	kErrTypeNoOneObject	

## Validation

3100	kErrWrongSymbolName	
3101	kErrWrongSymbol	
3102	kErrWrongStrLength	
3103	kErrDirDoesNotExist	
3104	kErrDirReadOnly	
3105	kErrAccessDenied	
3106	kErrFileDoesNotExist	
3107	kErrFileAlreadyExists	
3108	kErrFileOpenFailed	
3109	kErrFileReadOnly	
3110	kErrInvalidString	
3111	kErrOutOfRange	
3112	kErrWrongColor	
3113	kErrIncorrectParameters	One of ADE validation functions recognized incorrect input parameters. This error is an internal ADE error.
3114	kErrFileOpenLimit	
3115	kErrShareViolation	
3116	kErrNetAccessDenied	
3117	kErrPathDoesNotExist	

## File Locking

3200	kErrDwkFileDoesNotExist	ADE lock file is locked. Occurs when using ADEDRAWINGS and ADEQUERY commands and when ADE is running in a multi-user environment.
3201	kErrOpenDwkFileFailed	ADE was unable to open the .DWK lock file. Call support.
3202	kErrFileLockedByAcad	Attempt to remove a user who does not exist from the lock file. Call support.
3203	kErrOldMapLockFile	ADE was unable to create the .DWK lock file. Call support.
3204	kErrFileIsNotDwk	ADE tried to lock a file for write that was already locked for read. Call support.
3205	kErrSpecifiedUserDoesNotExist	ADE tried to lock a file for read that was already locked for write. Occurs during query operations in a multi-user environment.
3206	kErrCreateDwkFileFailed	ADE tried to open and read a file that was not a valid .DWK file. Call support.
3207	kErrFileIsLockedForRead	ADE was unable to unlock the lock file. Call support.
3208	kErrFileIsLockedForWrite	ADE tried to attach a file that is already open by AutoCAD. Occurs when using ADEDRAWINGS with ATTACH operations if the file is open in an AutoCAD project.
3209	kErrInvalidLockStateSpecified	ADE internal object is invalid. Call support.
3210	kErrNotOwnerOfWLH	ADE tried to remove a write lock when the user did not have a write lock. Call support.
3211	kErrUserIsNotWriter	
3212	kErrUserIsNotReader	
3213	kErrUserHasReadLock	
3214	kErrLockFileIsFull	
3215	kErrDwgFileDoesNotExist	ADE tried to unlock a file but the .DWK file was missing. Occurs if the .dwk file was erased after a file was attached.
3216	kErrNotAnADELockFile	ADE internal object is invalid. Call support.
3217	kErrFileMayHaveBeenModified	Existing .DWK file does not belong to ADE. the lock

		file exists and can be read by ADE, but ADE does not own the file.
3218	kErrFileHasLocks	ADE tried to remove a lock file but it was not found. Occurs if the .dwl file was erased after a file was attached.

### Unicode Support

4000	kErrUnicodeInsufficientBufferToConvert
4001	kErrUnicodeInvalidFlagsToConvert
4002	kErrUnicodeInvalidParameterToConvert
4003	kErrUnicodeNoTranslation
4004	kErrUnicodeCodePageNotAvailable

### Double-Byte Support

4005	kErrNoMBCSAllowed
------	-------------------

# AcMap::EErrType

[See Also](#)

AcMap::EErrType enumerators represent error types.

Errors on the error stack are represented by AcMapErrorEntry objects. Error types are returned by the AcMapErrorEntry::ErrorType function.

00	kAdeNoMessage	Error condition does not exist.
01	kAdeWarning	ADE (AutoCAD Data Extension) execution warning.
02	kAdeError	ADE execution error.
03	kAseWarning	ASE (AutoCAD SQL Extension) execution warning.
04	kAseError	ASE execution error.
05	kAcWarning	AutoCAD execution warning.
06	kAcError	AutoCAD execution error.
07	kAsiWarning	ASI (AutoCAD SQL Interface) execution warning.
08	kAsiError	ASI execution error.
09	kIRDWarning	Extended object data (Xdata) warning.
10	kIRDError	Xdata error.
13	kMapError	AutoCAD Map execution error.
14	kMentorError	
15	kApplicationError	Operating system level application error.
16	kDiagMessage	Diagnostic message returned.

# AcMap::EJoinOperator

Join operators for query conditions.

First	First	First.
2	kOperatorOr	

# AcMap::ELocationType

Location condition types.

First	First	First.
1	kLocationCrossing	

## AcMap::EOpenMode



0	kAdeClosed
1	kOpenForWrite
2	kOpenForRead

## AcMap::EPrefType

Option types.

0	kWSpaceType	Workspace.
1	kWSessType	Work session.
2	kQryType	Query.
3	kSvBkType	Save back.
4	kExtDbType	Database.
5	kCoordXformType	Coordinate transformation.
6	kSystemType	System.
7	kCoordsysType	Coordinate system.

# AcMap::EPreviewDefinitionsFrom

[See Also](#)

Source for preview definitions.

1	kProject
2	kSource
3	kLibrary

## AcMap::EProjectOptionType

Types of preferences.

00	kQryPrefDefn
01	kSvBkPrefDefn
02	kCoordXformPrefDefn

# AcMap::EPropertyType

Property condition types.

00	kArea
01	kBlockName
02	kColor
03	kElevation
04	kEntType
05	kGroup
06	kLayer
07	kLength
08	kLineType
09	kTextStyle
10	kTextValue
11	kThickness
12	kFeature
13	kLineweight
14	kPlotstyle

# AcMap::EQueryDialogOptions

Query dialog options.

00	kQueryDialogSuppressNothing
01	kQueryDialogSuppressExecute
02	kQueryDialogSuppressPreview
04	kQueryDialogSuppressReport
08	kQueryDialogSuppressAlteration
16	kQueryDialogSuppressDrawings
32	kQueryDialogSuppressMore

# AcMap::EQueryType

Query modes.

0	kQueryDraw
1	kQueryPreview
2	kQueryReport

# AcMap::ERangeOperator

Comparison operators for range lines.

1	kRangeEq	Equal to.
2	kRangeGT	Greater than.
3	kRangeGTorEq	Greater than or equal to.
4	kRangeLT	Less than.
5	kRangeLTorEq	Less than or equal to.
6	kRangeNotEq	Not equal to.
7	kRangeOtherwise	Otherwise.

# AcMap::ESaveQueryOptions

Save query options.

1	kSaveDrawingSet	Saves drawing objects.
2	kSaveLocationCoordinates	Saves location coordinates.
4	kSavePropertyAlteration	Saves property alterations.
8	kAutoExecute	Executes a query when it is loaded.

## AcMap::ETableType

Symbol table types.

0	kBlockTable	Block names.
1	kLayerTable	Layer names.
2	kLinetypeTable	Line types.
3	kStyleTable	Text styles.
4	kRegappTable	AutoCAD regapps.
5	kGroupTable	Group names.
6	kLinkTemplate	Link template names.
7	kODD	Object data table names.
8	kMLinestyleTable	Line styles.
9	kFeatureTable	Feature names.
10	kLineweightTable	Line weights.
11	kPlotstyleTable	Plot styles.

# AcMap::EUserRights

## User rights.

01	AcMap::kRightsSuperUser	Superuser
02	AcMap::kRightsAltDwgSet	Can alter drawing set
04	AcMap::kRightsEditDwg	Can edit drawings
08	AcMap::kRightsDrawQuery	Can run Draw queries
16	AcMap::kRightsAltClass	Can edit Feature Class definitions

## AcMap::SaveSetObjectType

Types of the objects in the save set.

1	kQueriedExisted	Queried objects that are unchanged or were altered by the query.
2	kQueriedErased	Queried objects that were deleted.
4	kNewlyCreated	New objects added to model space.



## Coordinate Transformation Functions

### [Data Extension Function Synopsis](#)

---

The coordinate transformation functions begin with `ade_proj`.

<a href="#">ade_projentitybackward</a>	Transforms an entity from the destination coordinate system to the source coordinate system.
<a href="#">ade_projentityforward</a>	Transforms an entity from the source coordinate system to the destination coordinate system.
<a href="#">ade_projgetctgname</a>	Identifies the category that a coordinate system belongs to.
<a href="#">ade_projgetinfo</a>	Gets information about a projection system.
<a href="#">ade_projgetwcode</a>	Gets the project drawing's coordinate system code.
<a href="#">ade_projlistctgy</a>	Lists available coordinate system categories.
<a href="#">ade_projlistcrdsysts</a>	Lists available coordinate systems in a given category.
<a href="#">ade_projptbackward</a>	Computes new coordinates for a source point.
<a href="#">ade_projptforward</a>	Computes new coordinates for a destination point.
<a href="#">ade_projsetdest</a>	Sets the destination coordinate system.
<a href="#">ade_projsetsrc</a>	Sets the source coordinate system.
<a href="#">ade_projsetwcode</a>	Sets the project drawing's coordinate system.
<a href="#">ade_projwsgeodistance</a>	Measures the geodetic distance between two points.



## ade\_aliasadd

### [Drive Alias Functions](#)

---

Creates a drive alias.

```
int  
ade_aliasadd(  
    char* alias_name,  
    char* path_name);
```

Returns **RTNORM** or an error code.

**alias\_name**      The name of the drive alias, which can be up to 31 characters. The name must be unique, contain no spaces, and start with an alphanumeric character.

**path\_name**      The drive and path to which the drive alias refers.

The function adds a new drive alias to the drive alias list. For example:

```
ade_aliasadd (  
    "ADSRX_SAMPLE",  
    "C:\\ADSRX\\Drawings\\PlotMapSet");
```



## ade\_aliasdelete

[Drive Alias Functions](#)

---

Deletes a drive alias.

int

```
ade_aliasdelete(  
    char* alias_name);
```

Returns **RTNORM** or an error code.

**alias\_name**      The name of the drive alias, which can be up to 31 characters. The name must be unique, contain no spaces, and start with an alphanumeric character.



## ade\_aliasgetlist

### [Drive Alias Functions](#)

---

Lists all drive aliases in the current drawing and all attached source drawings.

```
struct resbuf*  
ade_aliasgetlist();
```

Returns the list of drive aliases or **NULL**.

The following sample populates a **resbuf** with drive aliases using **ade\_aliasgetlist()**. The contents of the **resbuf** are displayed and the **resbuf** is released as required.

```
struct resbuf* pDriveAliasRb = ade_aliasgetlist();  
if(pDriveAliasRb != NULL) {  
    struct resbuf* rb = pDriveAliasRb;  
    while(rb != NULL) {  
        if (rb->restype == RTSTR) {  
            acutPrintf(  
                "\n\nThe \"%s\" alias contained the value:"  
                , rb->resval.rstring);  
            if (NULL != (rb = rb->rbnext)) {  
                acutPrintf(  
                    "\"%s\""  
                    , rb->resval.rstring);  
            }  
        }  
        rb = rb->rbnext;  
    }  
}  
else{  
    acutPrintf(  
        "\nNo alias could be determined.");  
}  
acutRelRb(pDriveAliasRb);
```



## ade\_aliasupdate

### [Drive Alias Functions](#)

---

Assigns a new drive and path to a drive alias.

```
int  
ade_aliasupdate(  
    char* alias_name,  
    char* path_name);
```

Returns **RTNORM** or an error code.

**alias\_name**      The name of the drive alias, which can be up to 31 characters. The name must be unique, contain no spaces, and start with an alphanumeric character.

**path\_name**      The new drive and path for the specified drive alias.



## ade\_altpclear

[Property Alteration Functions](#)

---

Clears the current property alteration definition.

```
int  
ade_altpclear();
```

Returns **RTNORM** or an error code.

A property alteration definition is a list of one or more property alteration expressions. See [ade\\_altpdefine](#) for information about property alteration expressions.



## ade\_altpdefine

### [Property Alteration Functions](#)

---

Creates a property alteration expression.

```
ade_id ade_altpdefine(  
    char* property,  
    struct resbuf* value);
```

Returns a property alteration expression ID or `ADE_NULLID`.

**property** Property to alter. See the Alterable Properties table below

**value** New value (type varies), or a range table expression that determines the new value. See [Using a Range Table](#) later in this topic.

**Note** If you use a range table expression instead of an explicit **value** argument, you must represent it in the **resbuf** as all one string.

You must release the **resbuf**.

### [Using a Range Table](#)

The following code generates a property alteration expression that adds a yellow text object to each queried entity. The text object displays which layer the entity came from. Note how the **value** argument is prepared by using `ads_buildlist` to define and populate the required **resbuf**.

```
char* propName = "textobject";  
struct resbuf* pPropValueRb = ads_buildlist  
(  
    RTLB,  
    RTLB,  
    RTSTR, "color", RTSTR, "yellow",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "textvalue", RTSTR, ".Layer",  
    RTDOT, RTDOT,  
    RTLE, 0  
);
```

```

ade_id propAltID = ade_altpdefine(propName, pPropValueRb);
if (ADE_NULLID != propAltID) {
    acutPrintf(
        "\nThe property alteration expression ID is: %.0lf"
        , propAltID);
}
else {
    acutPrintf(
        "\nNo property alteration expression ID was set");
}
acutRelRb(pPropValueRb);

```

A list of one or more property alteration expressions constitutes a property alteration definition. If there is a current property alteration definition when you create a property alteration expression, the new expression is added to it. When you execute a Draw query, each queried entity is altered in accord with the current property alteration definition.

The following table lists the alterable properties:

### Alterable Properties

<b>blockname</b>	Block name ( <b>RTSTR</b> )
<b>color</b>	<a href="#">Color</a> ( <b>RTSTR</b> )
<b>elevation</b>	Z coordinate ( <b>RTPOINT</b> ) in the user coordinate system
<b>height</b>	Text height ( <b>RTREAL</b> )
<b>layer</b>	Layer name ( <b>RTSTR</b> )
<b>linetype</b>	Line type ( <b>RTSTR</b> )
<b>rotation</b>	Rotation ( <b>RTREAL</b> )
<b>scale</b>	Scaling factor ( <b>RTREAL</b> ). For example, 1.2 = 120%
<b>style</b>	Text style ( <b>RTSTR</b> )
<b>width</b>	Line width ( <b>RTREAL</b> )
<b>textvalue</b>	Text value ( <b>RTSTR</b> )
<b>thickness</b>	Thickness ( <b>RTREAL</b> )
<b>hatch</b>	List of dotted pairs that define the hatch properties. See Hatch properties below

<b>textobject</b>	List of dotted pairs that define the text object properties. See Text object properties below
-------------------	--

To add a hatch pattern to each queried entity, as long as it is a closed polygon, specify "hatch" for the **property** argument. The **value** argument is then a list of dotted pairs. Each dotted pair is composed of a hatch property and a string value.

### Hatch Properties

<b>pattern</b>	Hatch pattern name (RTSTR)
<b>scale</b>	Scaling factor (RTSTR). For example, "1.2" = 120%
<b>rotation</b>	Rotation of the hatch pattern (RTSTR)
<b>layer</b>	Name of the layer that contains the hatch pattern (RTSTR)
<b>color</b>	Hatch pattern <a href="#">color</a> (RTSTR)

To create a text object for each queried entity, specify "textobject" for the **property** argument. The **value** argument is then a list of dotted pairs. Each dotted pair is composed of a text object property and a string value. The value element in the dotted pair can be an explicit value or a range table expression that determines a value.

### Text Object Properties

<b>textvalue</b>	Text to display (RTSTR)
<b>height</b>	Text height (RTSTR)
<b>inspt</b>	Point where text is inserted (expression as a RTSTR)
<b>justify</b>	Text alignment (RTSTR). For example, "center".
<b>style</b>	Text style (RTSTR)
<b>layer</b>	Name of the layer on which the text object resides (RTSTR)
<b>color</b>	Text <a href="#">color</a> (RTSTR)
<b>rotation</b>	Rotation of the text object (RTSTR)



## ade\_altpdelprop

[Property Alteration Functions](#)

---

Deletes a property alteration expression.

int

```
ade_altpdelprop(  
    ade_id altp_id);
```

Returns **RTNORM** or an error code.

*altp\_id*      Property alteration expression ID.



## ade\_altpgetprop

### [Property Alteration Functions](#)

---

Gets a property alteration expression.

```
struct resbuf*
ade_altpgetprop(
    ade_id altp_id);
```

Returns a property alteration expression ID or **NULL**.

[altp\\_id](#) Property alteration expression ID.

You must release the **resbuf**.

See [ade\\_altpdefine](#) for information about property alteration expressions.

The following example creates an initial property alteration expression using [ade\\_altpdefine\(\)](#). This expression is based on a "Text Object" type and sets the color property to yellow and the text property to .layer. The property alteration expression ID returned by [ade\\_altpdefine\(\)](#) is used by [ade\\_altpgetprop\(\)](#) and automatically assumes that a "Text Object" type is going to be accessed. The property names and the corresponding values associated with the "Text Object" type are displayed, then the **resbufs** are released as required.

```
char* propName = "textobject";
struct resbuf* pPropValueRb = acutBuildList(
    RTLB,
    RTSTR, "color",
    RTDOT,
    RTSTR, "yellow",
    RTLE,
    RTLB,
    RTSTR, "textvalue",
    RTDOT,
    RTSTR, ".Layer",
    RTLE, 0
);
ade_id propAltID = ade_altpdefine(propName, pPropValueRb);
```

```

if (ADE_NULLID != propAltID) {
    acutPrintf(
        "\n\nThe property alteration expression ID is: %.0lf"
        , propAltID);
}
else {
    acutPrintf(
        "\n\nNo property alteration expression ID was set");
}
acutRelRb(pPropValueRb);
struct resbuf* pPropAltExprRb = ade_altpgetprop(propAltID);
if (NULL != pPropAltExprRb) {
    acutPrintf(
        "\n\nThe \"%s\" property contains the following value pairs:"
        , pPropAltExprRb->resval.rstring);
    if (NULL != (pPropAltExprRb = pPropAltExprRb->rbnext)) {
        struct resbuf* rb = pPropAltExprRb;
        while(rb != NULL) {
            switch(rb->restype)
            {
                case RTSTR:
                    acutPrintf(
                        "\n\n\t\tThe \"%s\" property"
                        , rb->resval.rstring, rb->resval.rstring);
                    break;
                case RTDOTE:
                    if (NULL != (rb = rb->rbnext))
                        acutPrintf(
                            " contains the value \"%s\""
                            , rb->resval.rstring);
                    break;
                default:
                    break;
            }
            rb = rb->rbnext;
        }
    }
}
else {
    acutPrintf(
        "\n\nNo property alteration expression information was retrieved");
}
}

```

acutRelRb(pPropAltExprRb);



## ade\_altplist

### [Property Alteration Functions](#)

---

Lists the IDs of the current property alteration expressions.

```
struct resbuf*  
ade_altplist();
```

Returns a list of property alteration IDs or **NULL**.

This list of property alteration expressions returned by this function constitutes the current property alteration definition.

You must release the **resbuf**.

The following sample parses the **resbuf** returned by **ade\_altplist()** and prints a list of property expression ID's in the current project. Then it releases the **resbuf**, as required.

```
struct resbuf* pAltPropListRb = NULL;  
pAltPropListRb = ade_altplist();  
if (NULL != pAltPropListRb) {  
    int nAltProp = 0;  
    struct resbuf* rb = pAltPropListRb;  
    while(rb != NULL) {  
        ++nAltProp;  
        acutPrintf(  
            "\nThe property alteration expression id %d is: %.0lf"  
            ,nAltProp, rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no property alteration expressions in this project.");  
}  
acutRelRb(pAltPropListRb);
```



## ade\_altpsetprop

### [Property Alteration Functions](#)

---

Modifies a property alteration expression.

```
int  
ade_altpsetprop(  
    ade_id propId,  
    char* property,  
    struct resbuf* value);
```

Returns **RTNORM** or an error code.

<code>altp_id</code>	Property alteration expression ID.
<code>property</code>	Property to alter.
<code>value</code>	The new value (type varies).

See [ade\\_altpdefine](#) for information about properties and values.

The following example creates an initial property alteration expression using `ade_altpdefine()`. This expression is based on a "Text Object" type and sets the color property to yellow and the text property to .layer. The property alteration expression ID returned by `ade_altpdefine()` is used by `ade_altpsetprop()` and automatically assumes that a "Text Object" type is going to be modified. The property param identifies a specific property of the "Text Object" type and the value `resbuf` contains the new value for that specific property. If the operation is successful, information is displayed about the modification and the `resbuf(s)` are released as required.

```
char* pszPropertyType = "textobject";  
struct resbuf* pPropValuePairRb = acutBuildList(  
    RTLB,  
    RTSTR, "color",  
    RTDOT,  
    RTSTR, "yellow",  
    RTLE,  
    RTLB,  
    RTSTR, "textvalue",
```

```
        RTDOTE,  
        RTSTR, ".Layer",  
        RTLE, 0  
    );  
ade_id propAltID = ade_altpdefine(pszPropertyType, pPropValuePairRb);  
char* pszPropertyName = "color";  
struct resbuf* pNewPropValueRb = acutBuildList(RTSTR, "green", 0);  
int resultCode = ade_altpsetprop(propAltID, pszPropertyName, pNewPropValueRb);  
if (RTNORM == resultCode) {  
    acutPrintf(  
        "\nThe \"%s\" property list contains the property \"%s\" which was assigned a new value of \"%s\""  
        , pszPropertyType, pszPropertyName, pNewPropValueRb->resval.rstring);  
}  
else {  
    acutPrintf(  
        "\nNo property modification was performed.");  
}  
acutRelRb(pPropValuePairRb);  
acutRelRb(pNewPropValueRb);
```



## ade\_dsattach

### [Drawing Set Functions](#)

---

Attaches a drawing to the current drawing.

```
ade_id  
ade_dsattach(  
    char* dwgname);
```

Returns the ID of the drawing attached or **ADE\_NULLID**.

***dwgname***     The path alias and file name of the drawing to attach.

This function returns an ID even if the drawing does not exist. A system administrator can use this function to define a drawing set before the drawing files it references are created or installed. A drawing must exist before you can make it active.

```
ade_aliasadd ("ADSRX_SAMPLE", "C:\\ADSRX\\Drawings");  
ade_id dwgId101 = ade_dsattach("ADSRX_SAMPLE:\\101.dwg");  
if (ADE_NULLID == dwgId101) {  
    acutPrintf(  
        "\nThe specified drawing was not attached.");  
}  
else {  
    acutPrintf(  
        "\nThe specified drawing was successfully attached.");  
}
```



## ade\_dsdetach

### [Drawing Set Functions](#)

---

Detaches a drawing from the current drawing.

```
int  
ade_dsdetach(  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

**dwg\_id**     The drawing ID to detach.

The following sample detaches the drawing attached in the [ade\\_dsattach](#) sample:

```
int resultCode = ade_dsdetach("ADSRX_SAMPLE:\\101.dwg");  
if (RTNORM == resultCode) {  
    acutPrintf(  
        "\nThe specified drawing was detached.");  
}  
else {  
    acutPrintf(  
        "\nThe specified drawing was not detached.");  
}
```



## ade\_dsisnested

### [Drawing Set Functions](#)

---

Checks if a drawing has nested drawings.

```
int  
ade_dsisnested(  
    ade_id dwg_id);
```

Returns **ADE\_TRUE** if the drawing has drawings attached, or **ADE\_FALSE**.

**dwg\_id**     The drawing ID of the drawing to check.

If the drawing ID has the value **ADE\_NULLID**, the current drawing is checked to see if it has drawings attached.

Verify that the drawing in question is active before calling **ade\_dsisnested**. It is not possible to determine if an inactive drawing has nested drawings. If **dwg\_id** is not specified or is **NULL**, the function checks the current drawing to see if it has drawings attached.

The following sample checks the current project for attached drawings using **ade\_dslist()** and if they exist the attached drawing is checked for attached, (nested) drawings using **ade\_dsisnested()**. Status messages are displayed based on the result of the checks and the **resbuf** is released as required.

```
ade_boolean bAllNestedDwgs = ADE_FALSE;  
struct resbuf* pAttachedDwgListRb = ade_dslist(  
    ADE_NULLID,  
    bAllNestedDwgs);  
  
if (NULL != pAttachedDwgListRb) {  
    ade_id dwgId = pAttachedDwgListRb->resval.rreal;  
    int nNestedDwgs = ade_dsisnested(dwgId);  
    if (nNestedDwgs) {  
        acutPrintf(  
            "\n\nThe current project has an attached drawing which has a drawing attached to it.");  
    }  
} else {  
    acutPrintf(  

```

```
        "\nNo nested drawings were found.");
    }
}
else{
    acutPrintf(
        "\nNo drawings were attached to the current project.");
}
acutRelRb(pAttachedDwgListRb);
```



## ade\_dslist

### [Drawing Set Functions](#)

---

Lists the drawings attached to a specified drawing.

```
struct resbuf*
ade_dslist(
    ade_id dwg_id,
    ade_boolean nested);
```

Returns a list of drawing IDs or **NULL**.

**dwg\_id**      The drawing ID or **ADE\_NULLID**.

**nested**      Drawing is nested or not. Values can be: **ADE\_TRUE** or **ADE\_FALSE**.

If the **dwg\_id** argument is **ADE\_NULLID**, this function returns drawing IDs for the drawings attached to the current drawing.

If the **nested** argument is **ADE\_TRUE**, the list includes drawings that are directly attached and all nested drawings at every level below them. If the argument is **ADE\_FALSE**, the list includes only drawings that are directly attached.

You must release the **resbuf**.

The following sample parses the **resbuf** returned by **ade\_dslist()** and prints a list of attached drawing ID's. Then it releases the **resbuf**, as required.

```
struct resbuf* pAttachedDwgsRb = NULL;
pAttachedDwgsRb = ade_dslist(ADE_NULLID, 1);
if (NULL != pAttachedDwgsRb) {
    int nDwg = 0;
    struct resbuf* rb = pAttachedDwgsRb;
    while(rb != NULL) {
        ++nDwg;
        acutPrintf(
            "\nAttached drawing # %d has the ID of: %.0lf"
            ,nDwg, rb->resval.rreal);
        rb = rb->rbnext;
    }
}
```

```
    }  
  }  
  else {  
    acutPrintf(  
      "\nThere are no attached drawings in this project.");  
  }  
  acutRelRb(pAttachedDwgsRb);
```



## ade\_dsproplist

### [Drawing Set Functions](#)

---

Lists all values found in the drawing set for a given drawing property.

```
struct resbuf*  
ade_dsproplist(  
    char* property);
```

Returns a list of values or **NULL**.

**property**      Drawing property. See Drawing Properties below.

The function searches all active source drawings and returns a list of the values it finds for the given drawing property.

The following table shows property names and return values.

#### Drawing Properties

<b>object_type</b>	AutoCAD object types ( <b>string</b> )
<b>blockname</b>	Block names ( <b>string</b> )
<b>linetype</b>	Line type names ( <b>string</b> )
<b>textstyle</b>	Text style names ( <b>string</b> )
<b>attrib</b>	Attribute name tags ( <b>string</b> )
<b>extents</b>	Computed extents: the lower-left and upper-right points in the set of active source drawings. For example: ((2.20286 4.99866) (20.4689 12.3563))
<b>group</b>	Group names ( <b>string</b> )
<b>layer</b>	Layer names ( <b>string</b> )
<b>lpn</b>	Link templates ( <b>string</b> ). Note that link path names (LPNs) have been replaced by link templates in AutoCAD Map.

<b>objdata</b>	Names of object data tables. Table names can be up to 25 characters long ( <b>string</b> ). Must be unique, contain no spaces, and start with an alphanumeric character
<b>m linestyle</b>	Mline style ( <b>string</b> )
<b>feature</b>	Feature name ( <b>string</b> )
<b>lineweight</b>	Line weight ( <b>string</b> )
<b>plotstyle</b>	Plot style ( <b>string</b> )

The following sample parses the **resbuf** returned by **ade\_dsproplist()** and prints a list of object data tables. Then it releases the **resbuf**, as required.

```
char* pszPropValue = "objdata";
struct resbuf* pDwgSetPropertyRb = ade_dsproplist(pszPropValue);
if(NULL != pDwgSetPropertyRb ) {
    struct resbuf* rb = pDwgSetPropertyRb;
    while(rb != NULL) {
        acutPrintf(
            "\nThe current drawing set contains the following %s information: %s",
            pszPropValue, rb->resval.rstring);
        rb = rb->rbnext;
    }
}
else{
    acutPrintf(
        "\nNo data could be retrieved.");
}
acutRelRb(pDwgSetPropertyRb);
```



## ade\_dwgactivate

### [Drawing Functions](#)

---

Activates a drawing.

```
int  
ade_dwgactivate(  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

**dwg\_id**      Drawing ID.

You can attach a drawing that does not yet exist, but you cannot activate it. See [ade\\_dsattach](#).

The following sample gets a list, (**resbuf**) containing drawing IDs of drawings directly attached to the current project, (no nested drawings) using [ade\\_dslist\(\)](#). If attached drawings exist, the list is processed and each drawing in the list is deactivated using [ade\\_dwgdeactivate\(\)](#). The **resbuf** is then released as required.

```
struct resbuf* pAttachedDwgsRb = ade_dslist(ADE_NULLID, ADE_FALSE);  
int resultCode = RTERROR;  
if (NULL != pAttachedDwgsRb) {  
    struct resbuf* rb = pAttachedDwgsRb;  
    while(NULL != rb) {  
        resultCode = ade_dwgdeactivate(rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no attached drawings in this project.");  
}  
acutRelRb(pAttachedDwgsRb);
```



## ade\_dwgactualpath

[Drawing Functions](#)

---

Returns the actual path of a drawing.

char\*

```
ade_dwgactualpath(  
    ade_id dwg_id);
```

Returns the actual full path (without an alias) of the specified drawing or **NULL**. You must call the **free** function to deallocate the returned buffer.

*dwg\_id*      Drawing ID.



## ade\_dwgaliaspath

[Drawing Functions](#)

---

Returns the alias path of a drawing.

char\*

```
ade_dwgaliaspath(  
    const char* dwgPath);
```

Returns the drawing alias path or **NULL**. You must call the **free** function to deallocate the returned buffer.

**dwg\_path**      The actual path of the drawing.



## ade\_dwgattriblist

### [Drawing Functions](#)

---

Returns a list of attribute tags for the specified block name.

```
struct resbuf*  
ade_dwgattriblist(  
    ade_id dwg_id,  
    char* block_name);
```

Returns a list of attribute tags or **NULL**.

**dwg\_id**            The drawing ID of the attached drawing containing the "block\_name".  
**block\_name**        The name of the block for which to get attribute tags.

This function returns a list of the attribute tags, given a block name from the specified drawing.

The following sample retrieves the ID of an attached drawing using `ade_dwggetid()` then parses the `resbuf` returned by `ade_dwgattriblist()` and prints a list of block attributes. Then it releases the `resbufs`, as required.

```
char* pszDwgPathName = "MYDWGS:\\994049-2blk.dwg";  
ade_id dwgId = ade_dwggetid (pszDwgPathName);  
char* pszBlockName = "1BANK-TR";  
struct resbuf* pBlkAttrRb = ade_dwgattriblist(dwgId, pszBlockName);  
if(pBlkAttrRb != NULL) {  
    struct resbuf* rb = pBlkAttrRb;  
    acutPrintf(  
        "\nA block named %s was found in %s and contains the following attributes: \n\n",  
        pszBlockName, pszDwgPathName);  
    while(rb != NULL) {  
        acutPrintf(  
            "\n%s", rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else{  
    acutPrintf(  

```

```
    "\n %s was not found in %s",  
    pszBlockName, pszDwgPathName);  
}  
acutRelRb(pBlkAttrbRb);
```



## ade\_dwgdeactivate

### [Drawing Functions](#)

---

Deactivates a drawing.

```
int  
ade_dwgdeactivate(  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

**dwg\_id**      Drawing ID.

The following sample deactivates all the drawings in the drawing set.

```
struct resbuf* pAttachedDwgsRb = ade_dslist(ADE_NULLID, ADE_FALSE);  
int resultCode = RTERROR;  
if (NULL != pAttachedDwgsRb) {  
    struct resbuf* rb = pAttachedDwgsRb;  
    while(NULL != rb) {  
        resultCode = ade_dwgdeactivate(rb->resval.rreal);  
        rb = rb->rnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no attached drawings in this project.");  
}  
acutRelRb(pAttachedDwgsRb);
```



## ade\_dwggetid

### [Drawing Functions](#)

---

Gets the drawing ID of a drawing.

```
ade_id  
ade_dwggetid  
    char* dwgpathname);
```

Returns a drawing ID or **ADE\_NULLID**.

**dwg\_pathname**      The path alias and drawing file name.

The following sample creates a drive alias using **ade\_aliasadd()** then obtains a drawing id using **ade\_dwggetid()**. Information pertaining to the results of the operation is then displayed. Note, The drawing in question must be attached to the current project.

```
int resultCode = ade_aliasadd (  
    "ADSRX_SAMPLE",  
    "C:\\My Documents\\_MapAPI\\ADSRX\\Drawings");  
char* pszDwgPathName = "ADSRX_SAMPLE:\\ForQuery.dwg";  
ade_id dwgId = ade_dwggetid(pszDwgPathName);  
if (ADE_NULLID != dwgId) {  
    acutPrintf(  
        "\n\nThe specified drawing \"%s\" has a drawing ID of: %.0lf"  
        , pszDwgPathName, dwgId);  
    }  
else {  
    acutPrintf(  
        "\n\nNo ID could be obtained for the specified drawing.");  
    }  
}
```



## ade\_dwggetsetting

### [Drawing Functions](#)

---

Gets a drawing setting value.

```
struct resbuf*  
ade_dwggetsetting(  
    ade_id dwg_id,  
    char* setting);
```

Returns the value of the given drawing setting or **NULL**.

**dwg\_id**     Drawing ID.

**setting**     The drawing setting name. See Drawing Setting Names below

### Drawing Setting Names

Setting Name	Return Value
<b>dwgname</b>	Drawing name; a path alias and file name, such as "myfiles:\mydwg.dwg"
<b>dwgdesc</b>	Drawing description.
<b>t_scale</b>	Simple transform scale. For example, <b>1.2</b> = 120%
<b>t_rotate</b>	Simple transform rotation direction. Value depends on the AutoCAD <b>ANGDIR</b> setting ( <b>real</b> ). Values: 0 = counterclockwise 1 = clockwise
<b>t_xoffset</b>	Simple transform X offset.
<b>t_yoffset</b>	Simple transform Y offset.
<b>t_apply</b>	Flag value. Values: <b>1</b> = apply all simple transformations defined for the given drawing <b>0</b> = do not apply transformations
<b>saveback</b>	Save back coordinates, a sequence of corner points, in this order: lower left, lower

right, upper right, upper left, separated by "."

---

The following sample gets the drawing ID from an attached drawing using `ade_dwggetid()` then parses the `resbuf` returned by `ade_dwggetsetting()` and prints the information relevant to the setting specified. Then it releases the `resbuf`, as required.

```
ade_id dwgId;
char* pszDwgPathName = "MYDWGS:\\994049-2blk.dwg";
dwgId = ade_dwggetid( pszDwgPathName);

struct resbuf* pDwgSettingRb;
char* pszDwgSetting = "dwgdesc";
pDwgSettingRb = ade_dwggetsetting(dwgId, pszDwgSetting);
while(pDwgSettingRb != NULL) {
    acutPrintf(
        "\nThe %s information associated with attached drawing %s is: %s"
        ,pszDwgSetting, pszDwgPathName, pDwgSettingRb->resval.rstring);
    pDwgSettingRb = pDwgSettingRb->rbnext;
}
acutRelRb(pDwgSettingRb);
```



## ade\_dwghaslocks

[Drawing Functions](#)

---

Checks if a drawing has locked objects.

```
int  
ade_dwghaslocks(  
    ade_id dwg_id);
```

Returns **ADE\_TRUE** or **ADE\_FALSE**.

*dwg\_id*     Drawing ID.



## ade\_dwgindex

### [Drawing Functions](#)

---

Applies specified index operations to a drawing.

```
rtype  
ade_dwgindex(  
    ade_id dwgID);
```

Returns **RTNORM** or an error code.

**dwgID** The drawing ID for the drawing on which index operations will be performed.

When you call **ade\_dwgindex**, indexes in the specified drawing are created or removed in accordance with the current index operation definition created by one or more calls to **ade\_dwgindexdef**. If there is no index operation definition, **ade\_dwgindex** has no effect on the drawing and returns **ADE\_NULLID**.

The following sample, first clears any existing index definitions using **ade\_dwgindexdef()** with an empty string, ("") for the **indexType** parameter. A location index and an object data index are then defined and executed against all active drawings in the current project using **ade\_dwgindex()**. **Resbufs** are then released as required. For more samples, see [Creating Index Operation Expressions](#).

```
char* pszIndexName = "";  
struct resbuf* indexParams = acutBuildList(RTNIL, 0);  
int resultCode = ade_dwgindexdef(  
    pszIndexName, // An empty string "" clears  
    0, // remove index  
    indexParams);  
  
char* pszPropIndexName = "Location";  
resultCode = ade_dwgindexdef(  
    pszPropIndexName,  
    1,  
    indexParams);  
  
struct resbuf* pOdIndexParamsRb = acutBuildList(  
    RTLB,
```

```

        RTLB,
        RTSTR, "RIBLKGRP",
        RTSTR, "AMERI_ES",
        RTLE,
        RTLB,
        RTSTR, "RIRDS",
        RTLE,

        RTLE,
        0);
char* pszOdIndexName = "ObjData";
resultCode = ade_dwgindexdef(
    pszOdIndexName,
    1,
    pOdIndexParamsRb);

struct resbuf* pAttachedDwgsRb = ade_dslist(ADE_NULLID, ADE_FALSE);
if (NULL != pAttachedDwgsRb) {
    struct resbuf* rb = pAttachedDwgsRb;
    while(NULL != rb) {
        resultCode = ade_dwgindex(rb->resval.rreal);
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(
        "\nThere are no attached drawings in this project.");
}
acutRelRb(indexParams);
acutRelRb(pAttachedDwgsRb);

```



## ade\_dwgindexdef

### [Drawing Functions](#)

---

Specifies which indexes are to be created or removed.

```
int  
ade_dwgindexdef(  
    struct resbuf* indexType,  
    short indexOper,  
    struct resbuf* indexParams);
```

Returns **RTNORM** or an error code.

- |                    |   |
|--------------------|---|
| <b>indextype</b>   | Type of index to be operated on: "Location", "Property", "EED", "SQLLINKS", "ObjData".  |
| <b>indexoper</b>   | Index operator: <b>1</b> = create, <b>0</b> = remove.   |
| <b>indexparams</b> | Additional parameters required to specify which index is to be manipulated or <b>NULL</b> if none. Only the Object Data index supports additional parameters. |

The list of indexes to be created or removed and associated information constitutes an index operation expression. A list of one or more index operation expressions constitutes an index operation definition. If there is a current index operation definition when you create an index operation expression, the new expression is added to it. When you call **ade\_dwgindex**, indexes in the specified drawing are created or removed in accordance with the current index operation definition.

If you use "" (the empty string) for the **indexType** argument (in which case the arguments for the other two parameters do not matter so long as they are valid), the current index operation definition is cleared.

The **indexparams** argument is a list of tables and fields on which object data indexes will be created or removed (depending on the **indexoper** argument). The list consists of an object data table name followed by the fields to be processed. If this argument is **NULL**, all object data tables and fields will be processed.

The following sample gets the drawing ID from an attached drawing using **ade\_dwggetid()** then parses the **resbuf** used by **ade\_userlist()** and prints a status of the type of index created for the specified drawing. Then it releases the **resbuf**, as required. For more samples, see [Creating Index Operation Expressions](#).

```
char* pszDwgPathName = "MYDWGS:\\994049-2blk.dwg";  
ade_id dwgId = ade_dwggetid (pszDwgPathName);
```

```
int indexOper = 1;
struct resbuf* pDwgIndexOnRb = acutBuildList(
    RTSTR,"Property",
    RTSTR,"Location",
    RTSTR,"EED",
    0;
struct resbuf* rb = pDwgIndexOnRb;
while(rb != NULL) {
    int returnCode = ade_dwgindef(rb->resval.rstring, indexOper, NULL);
    if (RTNORM == returnCode) {
        acutPrintf(
            "\nThe %s index has been successfully created for %s"
            ,rb->resval.rstring, pszDwgPathName);
    }
    else {
        acutPrintf(
            "\nThe index could not be created.");
    }
    rb = rb->rbnext;
}
acutRelRb(pDwgIndexOnRb);
```



## ade\_dwgisactive

[Drawing Functions](#)

---

Checks if a drawing is active.

```
int  
ade_dwgisactive(  
    ade_id dwg_id);
```

Returns **ADE\_TRUE** or **ADE\_FALSE**.

**dwg\_id**      Drawing ID.

The function returns **ADE\_TRUE** if the specified drawing is active. If the drawing is not active or the drawing ID is invalid, the function returns **ADE\_FALSE**.



## ade\_dwgistoplevel

[Drawing Functions](#)

---

Checks if a drawing is directly attached to the current drawing.

int

```
ade_dwgistoplevel(  
    ade_id dwg_id);
```

Returns **ADE\_TRUE** if the drawing is attached directly to the current drawing, or **ADE\_FALSE**.

*dwg\_id*     Drawing ID.



## ade\_dwgproplist

### [Drawing Functions](#)

---

Lists all values found in a drawing for a given drawing property.

```
struct resbuf*  
ade_dwgproplist(  
    ade_id dwg_id,  
    char* property);
```

Returns a **resbuf** list or **NULL**.

**dwg\_id**      Drawing ID.  
**property**    Property name. See Property Names below.

You must release the **resbuf**.

The function searches the given drawing and returns a list of the values it finds for the given drawing property.

### Property Names

Name	Return Value.
<b>object_type</b>	AutoCAD object types.
<b>blockname</b>	Block names.
<b>linetype</b>	Line type names.
<b>textstyle</b>	Text style names.
<b>attrib</b>	Attribute tag names.
<b>extents</b>	Computed extents. The most lower-left point and the most upper-right point in the drawing For example: ((2.20286 4.99866) (20.4689 12.3563))
<b>group</b>	Group names.

<b>layer</b>	Layer names.
<b>lpn</b>	Link templates. Note that link path names (LPNs) have been replaced by link templates in AutoCAD Map
<b>objdata</b>	Names of object data tables.
<b>m linestyle</b>	Mline style.
<b>feature</b>	Feature name.
<b>lineweight</b>	Line weight.
<b>plotstyle</b>	Plot style.

The following sample gets a drawing ID from an attached drawing using `ade_dwggetid` then parses the `resbuf` returned by `ade_dwgproplist()` and prints a list of object data tables found in the attached drawing. Then it releases the `resbuf`, as required.

```
char* pszDwgPathName = "MYDWGS:\\SanFranMarina_BlkJrp_OD.dwg";
ade_id dwgId = ade_dwggetid (pszDwgPathName);
char* pszPropValue = "objdata";
struct resbuf* pDwgPropertyRb = ade_dwgproplist(dwgId, pszPropValue);
if(pDwgPropertyRb != NULL) {
    struct resbuf* rb = pDwgPropertyRb;
    while(rb != NULL) {
        acutPrintf(
            "\n\nThe current drawing set contains the following %s information: %s"
            ,pszPropValue, rb->resval.rstring);
        rb = rb->rbnext;
    }
}
else{
    acutPrintf(
        "\n\nNo information could be retrieved.");
}
acutRelRb(pDwgPropertyRb);
```



## ade\_dwgquickview

### [Drawing Functions](#)

---

Displays a quick view of a drawing.

```
int  
ade_dwgquickview(  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

**dwg\_id**      Drawing ID.

The following sample gets a list of drawings attached to the current project using **ade\_dslist()**. Then displays quick views of all attached drawings and releases the **resbuf** as required

```
struct resbuf* pAttachedDwgsRb = ade_dslist(ADE_NULLID, ADE_FALSE);  
int resultCode = RTERROR;  
if (NULL != pAttachedDwgsRb) {  
    struct resbuf* rb = pAttachedDwgsRb;  
    while(NULL != rb) {  
        resultCode = ade_dwgquickview(rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no attached drawings in this project.");  
}  
acutRelRb(pAttachedDwgsRb);
```



## ade\_dwgselectdlg

### [Drawing Functions](#)

---

Displays the Select Drawings dialog box.

```
struct resbuf*  
ade_dwgselectdlg(  
    void* pParent,  
    const char* pCaption);
```

Returns a **resbuf** list or **NULL**.

**pParent** A value defining the pointer to the Select Drawings dialog box parent window, which is expected to be represented by a CWnd object. 0 means that the parent window is not defined.

**pCaption** Text that is shown before the current directory in the caption of the Select Drawings dialog box.

You must release the **resbuf**.

The following sample populates a **resbuf** based on the files selected using **ade\_dwgselectdlg()**. The contents of the **resbuf** are displayed and then released as required.

```
void* pParentCWnd = 0;  
const char* pszDlgCaption = "Dialog Caption";  
struct resbuf* pSelectedDwgsRb = ade_dwgselectdlg(pParentCWnd, pszDlgCaption);  
if (NULL != pSelectedDwgsRb){  
    acutPrintf(  
        "\n\nThe following files were selected: ");  
    struct resbuf* rb = pSelectedDwgsRb;  
    while(NULL != rb) {  
        acutPrintf("\n\n\"%s\""  
            , rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else {
```

```
    acutPrintf(
        "\nNo files were selected.");
}
acutRelRb(pSelectedDwgsRb);
```



## ade\_dwgsetof

### [Drawing Functions](#)

---

Identifies the drawings to which a given drawing is attached.

```
struct resbuf*
ade_dwgsetof(
    ade_id dwg_id);
```

Returns a **resbuf** list or **NULL**.

**dwg\_id** Drawing ID.

You must release the **resbuf**.

You cannot use this function to check if a drawing is attached to the current drawing. Use [ade\\_dwgistoplevel](#) instead. If a drawing is attached to both the current drawing and to other drawings, this function returns a list of the IDs of the other drawings only.

The following sample returns the project name to which a specific drawing is attached.

**Ade\_dwggetid()** is used to obtain the drawing id of a file which is part of a drawing set, (BestRoute.dwg). A **resbuf** is populated with the name of the project which BestRoute.dwg is attached to using **ade\_dwgsetof()**. If the operation is successful, (the resbuf is not NULL) then the contents of that **resbuf** are displayed. The **resbuf** is then released as required.

```
char* pszDwgPathName = "ADSRX_SAMPLE:\BestRoute.dwg";
ade_id dwgId = ade_dwggetid(pszDwgPathName);
struct resbuf* pDwgAttachedToRb = ade_dwgsetof(dwgId);
if (NULL != pDwgAttachedToRb)
{
    struct resbuf* rb = pDwgAttachedToRb;
    while(rb != NULL)
    {
        char* pszDwgPath = ade_dwgactualpath(rb->resval.rreal);
        acutPrintf(
            "\nThe specified drawing \"%s\" is attached to: \n\n%s"
            , pszDwgPathName, pszDwgPath);
        rb = rb->rbnext;
    }
}
```

```
    }  
  }  
  else  
  {  
    acutPrintf(  
      "\nThe specified drawing is attached to no project.");  
  }  
  acutRelRb(pDwgAttachedToRb);
```



## ade\_dwgsetsetting

### [Drawing Functions](#)

---

Sets a drawing setting value.

```
int  
ade_dwgsetsetting(  
    ade_id dwg_id,  
    struct resbuf* proplist);
```

Returns **ADE\_TRUE** if the drawing is active or **ADE\_FALSE**.

**dwg\_id**      Drawing ID.

**proplist**     List composed of a setting name and a value. See Setting Names below.

### Setting Names

Setting name	Value
<b>dwgname</b>	Drawing name; a full path name, such as <i>c:\\drawings\\mydwg.dwg</i> .
<b>dwgdesc</b>	Drawing description.
<b>t_scale</b>	Simple transform scale. For example, <b>1.2</b> = 120%
<b>t_rotate</b>	Simple transform rotation; rotation direction depends on the AutoCAD <b>ANGDIR</b> setting.
<b>t_xoffset</b>	Simple transform X offset.
<b>t_yoffset</b>	Simple transform Y offset.
<b>t_apply</b>	Flag value. Values can be: <b>1</b> = apply all simple transformations defined for the given drawing <b>0</b> = do not apply transformations
<b>saveback</b>	Save back coordinates, a sequence of corner points, in this order: lower left, lower right, upper right, upper left, separated by "."

The following sample gets a drawing ID from an attached drawing using `ade_dwggetid()`. This ID and the `resbuf` containing the setting name/value pair are used by `ade_dwgsetsetting()` which returns a status code whose associated message is displayed. Then it releases the `resbuf`, as required.

```
ade_id dwgId = ade_dwggetid (pszDwgPathName);
struct resbuf* pDwgSettingsRb = ads_buildlist
    (
        RTLB,
        RTSTR, "dwgdesc",
        RTSTR, "This is a drawing description",
        RTDOT,
        0
    );
returnCode = ade_dwgsetsetting(dwgId, pDwgSettingsRb);
if (RTNORM == returnCode) {
    acutPrintf(
        "\nThe value has been successfully set for %s"
        ,pszDwgPathName);
}
else {
    acutPrintf(
        "\nThe value could not be set.");
}
acutRelRb(pDwgSettingsRb);
```



## ade\_dwgunlock

[Drawing Functions](#)

---

Removes all object locks from a drawing.

```
int  
ade_dwgunlock(  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

*dwg\_id*      Drawing ID.

Using this function requires superuser privileges.



## ade\_dwgzoomextents

[Drawing Functions](#)

---

Zooms to the extents of the active drawings.

```
int  
ade_dwgzoomextents();
```

Returns **RTNORM** or **RTERROR**.



## ade\_editdefcen

### [Object Editing Functions](#)

---

Defines a new label point for an object.

```
int  
ade_editdefcen(  
    ads_name ename,  
    ads_point pt);
```

Returns **RTNORM** or an error code.

**ename**      An AutoCAD entity name.  
**pt**          The label point, a list of values defined in 2D or 3D point.

Use this function with property alteration if the current label point is not suitable for the text object you are adding.

The following sample modifies the label point for every entity in the current drawing.

```
ads_name ename;  
ads_point newLabelPt;  
double newLabelPtOffset = 1.25;  
struct resbuf* pLabelPtValRb = NULL;  
ads_name selectionSet;  
acedSSGet("x", NULL, NULL, NULL, selectionSet);  
long ssLength;  
acedSSLength( selectionSet, &ssLength; );  
for(int i = 0; i < ssLength; ++i)  
{  
    if(acedSSName(selectionSet, i, ename) == RTNORM ) {  
        pLabelPtValRb = ade_expreval(  
            ename,  
            ".labelpt",  
            "point");  
        if (NULL != pLabelPtValRb) {  
            newLabelPt[X] = (pLabelPtValRb->resval.rpoint[0] - newLabelPtOffset);  
        }  
    }  
}
```

```
newLabelPt[Y] = (pLabelPtValRb->resval.rpoint[1] - newLabelPtOffset);
int resultCode = ade_editdefcen(
    ename,
    newLabelPt);
}
else {
    acutPrintf(
        "\nThe label point could not be determined.");
}
}
else {
    acutPrintf(
        "\nCould not get the entity.");
}
}
acutRelRb(pLabelPtValRb);
acedSSFree(selectionSet);
```



## ade\_editlockederased

[Object Editing Functions](#)

---

Gets the objects in the save set that have been erased.

int

```
ade_editlockederased(  
    ads_name sel_set);
```

Returns **RTNORM** or an error code

*sel\_set*     The name of the selection set.



## ade\_editislocked

### [Object Editing Functions](#)

---

Gets lock information about an object if it is locked.

```
struct resbuf*  
ade_editislocked(  
    ads_name ename);
```

Returns a **resbuf** list of lock information about the specified locked object, or, if it is not locked, **NULL**.

**ename**      AutoCAD entity name.

You must release the **resbuf**.

The list of lock information returned by this function contains the following, in order:

- Login name of the user who locked the object.
- Name and path of the drawing that contains the object.
- Date the object was locked.
- Time the object was locked.
- Name and path of the current drawing.

The following sample create a selection set of locked entities in a save back set using **ade\_editlocked()**. A **resbuf** is then populated with edit lock information for each of the entities in the selection set using **ade\_editislocked()**. That information is displayed and the **resbuf** is released as required.

```
ads_name selectionSet;  
int nResultCode = ade_editlocked(selectionSet);  
long ssLength;  
acedSSLength( selectionSet, &ssLength; );  
struct resbuf* pLockedEntitiesRb = NULL;  
for( int i = 0; i < ssLength; ++i )  
{  
    ads_name ename;  
    if( acedSSName( selectionSet, i, ename ) == RTNORM )  
    {
```

```

pLockedEntitiesRb = ade_editislocked(ename);
if (NULL != pLockedEntitiesRb) {
    acutPrintf(
        "\n\nThe locked entity contains the following information:");
    struct resbuf* rb = pLockedEntitiesRb;
    while(rb != NULL) {
        if (rb->restype == RTSTR) {
            acutPrintf(
                "\n\t\t\t\"%s\"""
                , rb->resval.rstring);
        }
        rb = rb->rbnext;
    }
}
}
}
}
acutRelRb(pLockedEntitiesRb);

```

You can change the format of the date and time strings through options in the International dialog box in the Microsoft Windows Control Panel.



## ade\_editlocked

[Object Editing Functions](#)

---

Gets the objects in the save set that have been modified or are new.

```
int  
ade_editlocked(  
    ads_name sel_set);
```

Returns **RTNORM** or an error code.

**sel\_set**     The name of the selection set.



## ade\_editlockobjs

### [Object Editing Functions](#)

---

Locks a set of objects and adds them to the save set.

```
ads_real  
ade_editlockobjs  
    ads_name sel_set);
```

Returns the number of objects locked or **ADE\_REALFAIL**.

**sel\_set**      The name of the selection set.

The function locks the objects contained in the designated selection set. Locking these objects adds them to the save set.

It is a good idea to compare the number of objects locked with the number of objects in the designated selection set. If the number locked is less than the number in the selection set, an error occurred in the locking process, and you should check the error stack.

The following sample creates a filtered selection set, adds its objects to a save set using **ade\_editlockobjs()**, checks the result and displays information based on those results. The **resbuf** and **selection set** is then released as required.

```
struct resbuf* pFilteredEntitySelectionRb = acutBuildList(  
    RTDXF0, "LWPOLYLINE",  
    8, "UtilityNetwork-Electric",  
    0);  
  
ads_name ssObjsForSaveSet;  
acedSSGet("X", NULL, NULL, pFilteredEntitySelectionRb, ssObjsForSaveSet);  
long filteredEntitySelectionLength = 0;  
acedSSLength(ssObjsForSaveSet, &filteredEntitySelectionLength);  
  
ads_real objsAddedToSaveSet = ade_editlockobjs(ssObjsForSaveSet);  
  
if (filteredEntitySelectionLength == objsAddedToSaveSet) {  
    acutPrintf(  
        "Number of objects locked: %d",  
        objsAddedToSaveSet);  
}
```

```
        "\nThe number of selected entities, (%d)"
        "matches the number added to the save set."
        , filteredEntitySelectionLength);
    }
else {
    acutPrintf(
        "\nThe number of selected entities, (%d)"
        "does not match the number added to the save set, (%d)."
        , filteredEntitySelectionLength, objsAddedToSaveSet);
    }
acutRelRb(pFilteredEntitySelectionRb);
int resultCode = acedSSFree(ssObjsForSaveSet);
```



## ade\_editnew

[Object Editing Functions](#)

---

Gets the objects in the saved set that are new.

```
int  
ade_editnew(  
    ads_name sel_set);
```

Returns **RTNORM** or an error code.

**sel\_set**     The name of the selection set.



## ade\_editunlockobjs

### [Object Editing Functions](#)

---

Unlocks a set of objects and removes them from the save set.

```
ads_real  
ade_editunlockobjs  
    ads_name sel_set);
```

Returns the number of objects unlocked. If for some reason the selection set is freed prior to the call to `ade_editunlockobjs()`, the number of number of objects unlocked is returned as 0.0.

`sel_set`      The name of the selection set.

The function unlocks the objects in the specified selection set. If the selection set is `ADE_REALFAIL`, the function unlocks all erased objects. Unlocking objects removes them from the save set.

The following sample is a continuation of the [ade\\_editlockobjs](#) sample.

```
ads_real objsRemovedFromSaveSet = ade_editunlockobjs(ssObjsForSaveSet);  
if (filteredEntitySelectionLength == objsRemovedFromSaveSet) {  
    acutPrintf(  
        "\nThe number of selected entities, (%d)"  
        " matches the number removed from the save set."  
        , filteredEntitySelectionLength);  
}  
else {  
    acutPrintf(  
        "\nThe number of selected entities, (%d)"  
        "does not match the number removed from the save set, (%d)."  
        , filteredEntitySelectionLength, objsRemovedFromSaveSet);  
}  
int resultCode = acedSSFree(ssObjsForSaveSet);
```



## ade\_entsetlocation

[Other Functions](#)

---

Sets a new text-label point for a drawing object.

```
int  
ade_entsetlocation(  
    ads_name ename,  
    ads_point pt);
```

Returns **RTNORM** or an error code.

**ename**      An AutoCAD entity name.

**pt**          The new text-label point, a 2D or 3D point.

An drawing object's text-label point is the starting position for text added during a query property alteration. By default, the centroid of the object is the text-label point.



## ade\_errclear

[Error Message Functions](#)

---

Clears the error stack.

```
int  
ade_errclear()
```

Returns **RTNORM** or an error code.



## ade\_errcode

[Error Message Functions](#)

---

Gets the error code for a given error in the stack.

```
int  
ade_errcode(  
    int err_index);
```

Returns an error code or **NULL**.

**err\_index**      The position of the error in the stack, where  
                  **0** = first error

For a list of error codes, see [Error Codes](#).



## ade\_errgetlevel

[Error Message Functions](#)

---

Gets the system error level.

```
int  
ade_errgetlevel();
```

Returns an error level or an error code.



## ade\_errmsg

### [Error Message Functions](#)

---

Gets the error message for a given error in the stack .

char\*

```
ade_errmsg(  
    int err_index);
```

Returns an error message or **NULL**.

**err\_index**      Position of the error in the stack, where  
                  **0** = first error.



## ade\_errpush

### [Error Message Functions](#)

---

Pushes an error to the stack .

```
int  
ade_errpush(  
    int err_code,  
    char* level,  
    char* message);
```

Returns **RTNORM** or an error code.

**err\_code**      Error code (details below).

**level**          Error level. Values can be: "**warning**", "**error**" (default), or the empty string. If the empty string, the error level is "**error**".

**message**        Error message.

If your application will use custom error codes, define a range for them that does not conflict with any range used for AutoCAD Map [error codes](#). To specify a general error, let the **err\_code** argument be **1** (kAdeErr).

The valid **level** values, "**warning**", and "**error**", correspond respectively to the following error types:

1	kAdeWarning
2	kAdeError

For a list of all error types, including the two that are valid **level** values, see [Error Types](#).

The following example pushes an error to the stack.

```
ade_errpush( 1, "error", "message text" );
```



## ade\_errpushstatement

### [Error Message Functions](#)

---

Pushes a faulty SQL statement to the error stack.

```
int  
ade_errpushstatement(  
    char* statement,  
    int position);
```

Returns **RTNORM** or an error code.

**statement**      Faulty statement that caused the error.

**position**      Starting position of the error in the faulty statement. Position 1 corresponds to the first character.

This function is designed to add diagnostic information to an error you have just pushed. It is associated with the latest error in the stack only. A call to **ade\_errpushstatement** makes sense only if a call to [ade\\_errpush](#) immediately precedes it.



## ade\_errqty

[Error Message Functions](#)

---

Returns the number of errors in the stack.

```
int  
ade_errqty();
```

Returns an error count or **ADEERROR**.



## ade\_errsetlevel

[Error Message Functions](#)

---

Sets the system error level.

```
int  
ade_errsetlevel(  
    int level);
```

Returns **RTNORM** or an error code.

**level**      Error level. Values can be: **0**, **1**, or **2**

See [ade\\_errgetlevel](#) for details.



## ade\_errshowdlg

[Error Message Functions](#)

---

Displays the Map Messages dialog box.

```
int  
ade_errshowdlg();
```

Returns **RTNORM** or an error code.



## ade\_errstatement

### [Error Message Functions](#)

---

Gets the faulty SQL statement for a given error in the stack .

```
struct resbuf*  
ade_errstatement(  
    int err_index);
```

Returns a statement (*string*) and a starting position (*short*), or **NULL**

*err\_index*      Position of the error in the stack  
                  0 = first error

The SQL statement and error position are returned in the following format.

```
(faulty_statement err_pos)
```

The *faulty\_statement* string quotes the faulty SQL statement that caused the error. The *err\_pos* value identifies the starting position of the error in the faulty statement. Position **1** is the first character of the statement.

The expression (*ade\_errstatement 2*), which references the third error in the stack, could return

```
("xxx" 6)
```

where "xxx" is the faulty statement and **6** tells you that the trouble begins at the sixth character.

You must release the *resbuf*.

The following sample concludes by printing a faulty SQL statement to the screen.

The first step is to define a SQL query containing a faulty SQL statement which is in a *resbuf*. *Ade\_qrydefine()* is then called with all required parameters. The error stack is cleared using *ade\_errclear()* and the query is executed using *ade\_qryexecute()*. The error stack is checked for any new errors using *ade\_errqty()*, and if they exist a *resbuf* is populated with the SQL Select statement and the position that the syntactic error occurred. The contents of the *resbuf* are displayed and all *resbufs* are released as required.

```
char* pszJoinOperator = ""; // none  
char* pszBgnCondGrouping = ""; // none
```

```

char* pszNotOperator = ""; // none
char* pszCondType = "SQL";
char* pszEndCondGrouping = ""; // none

struct resbuf* pQueryConditionRb = acutBuildList(
    RTLB,
    RTSTR, "BlockGroup",
    RTSTR, "AMERI_ES == 0",
    RTLE,
    0);
ade_id queryId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pQueryConditionRb,
    pszEndCondGrouping);
int returnCode = ade_errclear();
ads_real queriedObjs = ade_qryexecute();
int errorStkQnty = ade_errqty();
if (errorStkQnty > 0)
{
    struct resbuf* pErrStatementRb = ade_errstatement(0);
    if (NULL != pErrStatementRb){
        struct resbuf* rb = pErrStatementRb;
        while(NULL != rb) {
            if (rb->restype == RTSTR) {
                if (_tcslen(rb->resval.rstring) != 0)
                {
                    acutPrintf("\n\nThe faulty SQL statement is: \n\n\t\"%s\"
                        , rb->resval.rstring);

                    acutPrintf("\n\n\t\tThe syntax error has occurred at character: %d"
                        , rb->rbnext->resval.rint);
                }
            }
            rb = rb->rbnext;
        }
    }
}
else {
    acutPrintf(
        "\nNo error statements were returned.");
}

```

```
    }  
    acutRelRb(pErrStatementRb);  
  }  
  else  
  {  
    acutPrintf("\nNo errors were encountered.");  
  }  
  acutRelRb(pQueryConditionRb);
```



## ade\_errtype

[Error Message Functions](#)

---

Gets the type of a given error in the stack.

```
int  
ade_errtype(  
    int err_index);
```

Returns an [error type](#) or `ADEERROR`.

`err_index`      Position of the error in the stack  
                  0 = first error

For a list of error types, see [Error Types](#).



## ade\_expreval

### [Expression Evaluation Functions](#)

---

Evaluates an expression.

```
struct resbuf*
ade_expreval(
    ads_name ename,
    char* expr,
    char* type);
```

Returns the value of the expression or **NULL**.

- ename** Optional drawing object name. Required if the expression uses object properties or data.
- expr** Expression to evaluate.
- type** Expected return type: "short", "long", "real", "string", or "point".

The following sample displays the .Length property of a pline entity which has been converted to Meters.

A selection set of entities from the current drawing is created using [acedSSGet\(\)](#). Each entity is passed to [ade\\_expreval\(\)](#) along with any additional parameters. The **resbuf** returned by [ade\\_expreval\(\)](#) contains the converted result which is then displayed. Then it releases the **resbuf**, as required.

```
ads_name ename;
ads_name selectionSet;
char* pszExpr = "(* .LENGTH .3048)";
char* pszReturnType = "real";
struct resbuf* pEntLengthToM = NULL;
acedSSGet("x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        pEntLengthToM = ade_expreval(ename, pszExpr, pszReturnType);
    }
}
```

```
if (pEntLengthToM != NULL){
    struct resbuf* rb = pEntLengthToM;
    while(rb != NULL) {
        acutPrintf(
            "\nThe current entities .LENGTH value converted to Meters is: %.4lf"
            , rb->resval.rreal);
        rb = rb->rbnext;
    }
}
}
}
}
acutRelRb(pEntLengthToM);
```



## ade\_keycolumnlist

[SQL Environment Functions](#)

---

Returns a list of the key column names for the specified link template.

```
struct resbuf*  
ade_keycolumnlist(  
    char* linktemplate);
```

Returns a list of key column names or **NULL**.

**linktemplate**      Link template.

For more information about link templates and using SQL, see the AutoCAD online documentation.

The following sample populates a **resbuf** with key column names using **ade\_keycolumnlist()**. The contents of the **resbuf** are displayed and the **resbuf** is released as required.

```
char* pszLinkTemplate = "BlockGroup";  
struct resbuf* pKeyColumnListRb = ade_keycolumnlist(pszLinkTemplate);  
if (NULL != pKeyColumnListRb) {  
    struct resbuf* rb = pKeyColumnListRb;  
    while(rb != NULL) {  
        acutPrintf(  
            "\nThe Key column name(s) are: %s"  
            , rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no specified link templates in this project.");  
}  
acutRelRb(pKeyColumnListRb);
```



## ade\_odaddfield

### [Object Data Functions](#)

---

Adds fields to a table.

```
int
ade_odaddfield(
    char* tablename,
    struct resbuf* fieldlist);
```

Returns **RTNORM** or an error code.

**tablename** The table name, which can be up to 25 characters long. It must be unique, contain no spaces, and start with an alphanumeric character.

**fieldlist** The list of fields to add; a sequence of field definitions.

A sequence of field definitions is introduced by the string "**columns**". Each field definition is a list of a-lists, and each a-list consists of a field property and a value, as follows:

Field property	Value
<b>colname</b>	Field name ( <b>RTSTR</b> ) can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>coldesc</b>	Field description ( <b>RTSTR</b> )
<b>coltype</b>	<a href="#">Field data type</a>
<b>defaultval</b>	Default field value

The function adds fields to the table and to each of its records. In each record, the new fields are assigned default values in accord with their field definitions. The function has no effect on existing fields. In other words, the function adds fields to each set of object data defined by the table and attached to an object.

You must release the **resbuf**.

For an example, see [Adding Fields to a Table](#).

The following sample creates a **resbuf** containing specifications for a new field, "field1" in the existing

object data table, "table1". This **resbuf** is passed to **ade\_odaddfield()** which returns a result code indicating the success of the operation. Then it releases the **resbuf**, as required.

```
struct resbuf* pAddFieldsRb = NULL;
char* pszOdTable = "table1";
pAddFieldsRb = acutBuildList(
    RTSTR, "columns",
    RTLB,
    RTLB,
    RTSTR, "colname", RTSTR, "field1",
    RTDOTE,
    RTLB,
    RTSTR, "coldesc", RTSTR, "Field 1 Description",
    RTDOTE,
    RTLB,
    RTSTR, "coltype", RTSTR, "character",
    RTDOTE,
    RTLB,
    RTSTR, "defaultval", RTSTR, "Default Value",
    RTDOTE,
    RTLE,
    // Define more fields as needed
    0);
int nResultCode = ade_odaddfield(pszOdTable, pAddFieldsRb);
acutRelRb(pAddFieldsRb);
if (RTNORM == nResultCode) {
    acutPrintf(
        "\nThe specified field has been successfully added.");
}
else {
    acutPrintf(
        "\nThe specified field has not been added.");
}
acutRelRb(pAddFieldsRb);
```



## ade\_odaddrecord

### [Object Data Functions](#)

---

Attaches data to an object.

```
int  
ade_odaddrecord(  
    ads_name ename,  
    char* table);
```

Returns **RTNORM** or an error code.

- ename**      An AutoCAD object name.
- table**      The table name, which can be up to 25 characters long. It must be unique, contain no spaces, and start with an alphanumeric character.

Attaching data to an object is also called attaching a table to an object. This function attaches a new record in a specific table to a specific object. Typically, a record contains information about whatever it is that the object represents. For example, if a line in a drawing represents a section of pipe in a water system, an attached record could contain information about that section.

When a new record is attached, its fields contain default values that correspond to their field definitions. To get a field value, use [ade\\_odgetfield](#); to change it, use [ade\\_odsetfield](#). Field definitions are included in the table definition. See [ade\\_oddefinetab](#) for information about table definitions.

You can attach more than one record to the same object with additional calls to **ade\_odaddrecord**. The additional records can be members of the same or different tables. If an object has only one record from a given table, the number of that record is 0. If you attach a second record from the same table, the number of that record is 1, and so on. Use [ade\\_odrecordqty](#) to find how many records of a given table are attached.

For example, if a section of water pipe is inspected at intervals, you could attach a number of records of the WATER INSPECTION table to the same line in the WATER drawing, and each record could contain the result of a different inspection.

The following sample creates a selection set of all entities in the current drawing using **acedSSGet()**. Each entity is then passed to **ade\_odaddrecord()** with all required parameters. The result code returned by **ade\_odaddrecord()** is displayed. Then it releases the **resbuf**, as required.

```
ads_name selectionSet;
```

```
ads_name ename;
char* pszOdTable = "table1";
acedSSGet("x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        int nResultCode = ade_odaddrecord(ename, pszOdTable);

        if (RTNORM == nResultCode) {
            acutPrintf(
                "\nThe specified record has been added.");
        }
        else {
            acutPrintf(
                "\nThe specified record has not been added.");
        }
    }
}
```



## ade\_odattachrecord

[Object Data Functions](#)

---

Attaches a new record to an object.

```
int  
ade_odattachrecord(  
    ads_name ename,  
    ade_id rec_id);
```

Returns **RTNORM** or an error code.

**ename**      An AutoCAD object name.

**rec\_id**     Record ID returned by **ade\_odnewrecord**.



## ade\_oddefinetab

### [Object Data Functions](#)

---

Creates an object data table.

```
int  
ade_oddefinetab(  
    struct resbuf* tab_defn);
```

Returns **RTNORM** or an error code

**tab\_defn** List of table elements: the table name, the table description, and a sequence field definitions.

The following sample creates a **resbuf** containing table definition parameters. **ade\_oddefinetab()** uses this **resbuf** and returns a result code indicating the success of the operation. Then it releases the **resbuf**, as required.

```
struct resbuf* pOdTableDefRb = acutBuildList(  
    RTLB,  
    RTSTR, "tablename", RTSTR, "NEWTABLE",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "tabledesc", RTSTR, "New Sample Table",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "columns",  
    RTLB,  
    RTLB,  
    RTSTR, "colname", RTSTR, "FIELD1",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "coldesc", RTSTR, "Field 1 Description",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "coltype", RTSTR, "character",  
    RTDOT, RTDOT,
```

```
    RTLB,  
        RTSTR, "defaultval", RTSTR, "Default Value",  
    RTDOTE,  
    RTLE,  
    // Define more fields as needed  
    RTLE,  
    0);  
int nResultCode = ade_oddefinetab(pOdTableDefRb);  
if (RTNORM == nResultCode) {  
    acutPrintf(  
        "\nThe specified table has been successfully created.");  
}  
else {  
    acutPrintf(  
        "\nThe specified table has not been created.");  
}  
acutRelRb(pOdTableDefRb);
```



## ade\_oddeletefield

### [Object Data Functions](#)

---

Deletes fields from a table.

```
int  
ade_oddeletefield(  
    char* tablename,  
    struct resbuf* fieldList);
```

Returns **RTNORM** or an error code.

**tablename**     The table name, which can be up to 25 characters long. It must be unique, contain no spaces, and start with an alphanumeric character.

**fieldlist**     A list of field names.

The function deletes the fields from the table and from each of its records. The data contained in these fields is also deleted. In other words, it deletes the fields and their data from each set of object data defined by the table and attached to an object.

**Note** The **fieldlist** argument for **ade\_oddeletefield** is a list of field names only. In the companion functions, **ade\_odaddfield** and **ade\_odmodifyfield**, it is a list of field definitions.

The **ade\_oddeletefield** function affects all active drawings in the drawing set. There should not be any queried objects for this operation.

**Note** This function will not operate unless your end user has superuser privileges.

The following sample creates a **resbuf** containing the fields to delete. **ade\_oddeletefield()** uses this **resbuf** and returns a result code indicating a successful operation. Then it releases the **resbuf**, as required.

```
struct resbuf* pDeleteFieldsRb = NULL;  
char* pszOdTable = "table1";  
pDeleteFieldsRb = acutBuildList(  
    RTSTR, "field1",  
    RTSTR, "field2",  
    RTSTR, "field3",  
    0 );  
int nResultCode = ade_oddeletefield( pszOdTable, pDeleteFieldsRb);
```

```
if (RTNORM == nResultCode) {
    acutPrintf(
        "\nThe fields specified have been successfully deleted.");
}
else {
    acutPrintf(
        "\nThe fields specified have not been deleted.");
}
acutRelRb(pDeleteFieldsRb);
```



## ade\_oddeletetab

[Object Data Functions](#)

---

Deletes a table.

```
int  
ade_oddeletetab(  
    char* tablename);
```

Returns **RTNORM** or an error code.

**tablename** The table name, which can be up to 25 characters long. It must be unique, contain no spaces, and start with an alphanumeric character.

The function deletes a table and all of its records. It deletes every set of object data defined by the table and attached to an object, as well as the data contained in the records.

The **ade\_oddeletetab** function affects all active drawings in the drawing set. There should not be any queried objects for this operation.

**Note** This function will not operate unless your end user has superuser privileges.



## ade\_oddelrecord

### [Object Data Functions](#)

---

Deletes a record.

```
int  
ade_oddelrecord(  
    ads_name ename,  
    char* table,  
    int recnum);
```

Returns **RTNORM** or an error code.

- ename**      The AutoCAD entity name of the object to which the record is attached.
- table**      The name of the table to which the record belongs, up to 25 characters long. Must be unique, contain no spaces, and start with a character.
- recnum**     The record number. The number of the first record is **0**.

The function deletes the record from the object. It deletes the set of object data defined by the table and attached to the object. This deletes the record from the table as well as the data contained in the record.

The record number is necessary because more than one record from the same table can be attached to an object. Use [ade\\_odrecordqty](#) to find how many records of a given table are attached.



## ade\_odfreerec

[Object Data Functions](#)

---

Frees the memory claimed in defining a new record.

```
int  
ade_odfreerec(  
    ade_id rec_id);
```

Returns **RTNORM** or an error code.

**rec\_id**     The record ID returned by **ade\_odnewrecord**.

**Warning** You must release a new record when you are finished with it.



## ade\_odgetfield

### [Object Data Functions](#)

---

Gets a field value.

```
struct resbuf*
ade_odgetfield(
    ads_name ename,
    char* table,
    char* field,
    int recnum);
```

Returns a field value or **NULL**.

<b>ename</b>	AutoCAD object name.
<b>table</b>	Table name can be up to 25 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>field</b>	Field name can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>recnum</b>	Record number. The number of the first record is <b>0</b> .

To identify a unique record, you need to specify the table to which it belongs, the object to which it is attached, and its record number. The record number is necessary because more than one record from the same table can be attached to an object. For more information about records and record numbers, see [ade\\_odaddrrecord](#).

The field value returned can be one of four [data types](#): **integer**, **character**, **point**, or **real**.

The following sample creates a selection set of all entities in the current drawing using `acedSSGet()`. Each entity in the selection set is passed to `ade_odgetfield()` with all required parameters, the results are stored in a **resbuf** which is displayed. Then it releases the **resbuf**, as required.

```
struct resbuf* pGetFieldsRb = NULL;
ads_name selectionSet;
ads_name ename;
char* pszOdTable = "table1";
char* pszOdField = "field1";
```

```
int recnum = 0;
acedSSGet("x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        pGetFieldsRb = ade_odgetfield(ename, pszOdTable, pszOdField, recnum);
        struct resbuf* rb = pGetFieldsRb;
        while(rb != NULL) {
            acutPrintf(
                "\n\nThe current entity has %s object data with a %s value of: %s"
                , pszOdTable, pszOdField, rb->resval.rstring);
            rb = rb->rbnext;
        }
    }
}
acutRelRb(pGetFieldsRb);
```



## ade\_odgetrecfield

### [Object Data Functions](#)

---

Gets a field value using a record ID.

```
struct resbuf*  
ade_odgetrecfield(  
    ade_id recID,  
    char* field);
```

Returns a field value (type varies) or **NULL**.

**recID**      Record ID returned by **ade\_odgetrecord**.  
**field**      Field name.

This function uses the record ID assigned by **ade\_odgetrecord** to get the value of a particular field. This means of getting an object data field value is generally faster than any other.

You must release the **resbuf**.

The following sample creates a selection set of all entities in the current drawing using **acedSSGet()**. Each entity in the selection set is passed to **ade\_odgetrecord()** which returns a record id. This record id along with other required parameters, are passed to **ade\_odgetrecfield()** which stores its results in a **resbuf** which is displayed. Then it releases the **resbuf**, as required.

```
struct resbuf* pOdRecFieldRb = NULL;  
ads_name selectionSet;  
ads_name ename;  
char* pszOdTable = "table1";  
char* pszOdField = "FIELD1";  
int recnum = 0;  
acedSSGet("x", NULL, NULL, NULL, selectionSet);  
long ssLength;  
acedSSLength( selectionSet, &ssLength; );  
ade_id recordId;  
for( int i = 0; i < ssLength; ++i )  
{
```

```
if( acedSSName(selectionSet, i, ename) == RTNORM )
{
    recordId = ade_odgetrecord(ename, pszOdTable, recnum);
    pOdRecFieldRb = ade_odgetrecfield(recordId, pszOdField);
    struct resbuf* rb = pOdRecFieldRb;
    while(rb != NULL) {
        acutPrintf(
            "\nThe current entity has %s object data with a %s value of: %s"
            , pszOdTable, pszOdField, rb->resval.rstring);
        rb = rb->rbnext;
    }
}
}
acutRelRb(pOdRecFieldRb);
```



## ade\_odgetrecord

### [Object Data Functions](#)

---

Gets a record ID.

```
ade_id  
ade_odgetrecord(  
    ads_name ename,  
    char* table,  
    int recnum);
```

Returns a record ID or **ADE\_NULLID**.

**ename**      AutoCAD object name.

**table**      Table name.

**recnum**      Record number; the first record number is **0**.

The function assigns an ID to the record uniquely determined by the three arguments. Later you can use this record ID with **ade\_odgetrecfield** to return the value of a particular field of this record. This means of getting an object data field value is generally faster than any other.

Three arguments are necessary because an AutoCAD object can be associated with more than one record in a table, in which case the records are distinguished by their record numbers. If there is only one record, its number is **0**. For more information about records and record numbers, see [ade\\_odaddrrecord](#).

The following sample creates a selection set of all entities in the current drawing using **acedSSGet()**. Each entity in the selection set is passed to **ade\_odgetrecord()** with all required parameters, the results are checked for value and displayed.

```
ads_name selectionSet;  
ads_name ename;  
char* pszOdTable = "Table1";  
int recnum = 0;  
acedSSGet("_x", NULL, NULL, NULL, selectionSet);  
long ssLength;  
acedSSLength( selectionSet, &ssLength; );  
ade_id recordId;
```

```
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        recordId = ade_odgetrecord(ename, pszOdTable, recnum);
        if (0 == recordId) {
            acutPrintf(
                "\nNo record id could be obtained.");
        }
        else {
            acutPrintf(
                "\nade_odgetrecord() returned: %.0lf"
                , recordId);
        }
    }
}
}
```



## ade\_odgettables

### [Object Data Functions](#)

---

Lists the tables attached to an object.

```
struct resbuf*
ade_odgettables(
    ads_name ename);
```

Returns a list of table names ([string](#)) or **NULL**.

**ename**      AutoCAD object name.

You must release the **resbuf**.

An object can have records of more than one table attached. This function lists all the tables that have records attached to the object. See [ade\\_odaddrecord](#) for information about records attached to objects.

An object can have more than one record from the same table attached. To find how many records of a given table are attached, use [ade\\_odrecordqty](#).

The following sample creates a selection set of all entities in the current drawing using **acedSSGet()**. Each entity in the selection set is passed to **ade\_odgettables()** with all required parameters, the results are stored in a **resbuf** and displayed. Then it releases the **resbuf**, as required.

```
struct resbuf* pOdTablesRb = NULL;
ads_name selectionSet;
ads_name ename;
acedSSGet("x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        pOdTablesRb = ade_odgettables(ename);
        struct resbuf* rb = pOdTablesRb;
        while(rb != NULL) {
            acutPrintf(
```

```
        "\nThe current entity has object data from %s attached: "  
        , rb->resval.rstring);  
    rb = rb->rbnext;  
    }  
    }  
    }  
acutRelRb(pOdTablesRb);
```



# ade\_odmodifyfield

## [Object Data Functions](#)

---

Modifies field properties in a table.

```
int  
ade_odmodifyfield(  
    char* tableName,  
    struct resbuf* fieldList);
```

Returns **RTNORM** or an error code.

**tableName** Table name can be up to 25 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.

**fieldList** Fields to modify. A sequence of field definitions. See Field Definitions below.

The **ade\_odmodifyfield** function affects all active drawings in the drawing set. There should not be any queried objects for this operation.

**Note** This function will not operate unless your end user has superuser privileges.

### Field Definitions

The field definitions are introduced by the "**columns**" string. Each field definition is a list of a-lists, and each a-list consists of a field property and a value, as follows:

Field Property	Field Value
<b>colname</b>	Field name ( <b>RTSTR</b> ) can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character
<b>coldesc</b>	Field description ( <b>RTSTR</b> )
<b>coltype</b>	<a href="#">Field data type</a>
<b>defaultval</b>	Default field value

In the **fieldlist** argument, specify the fields to modify and their new field definitions. The function acts on

these fields only and has no effect on any others. For each field you specify, the function replaces the existing field definition with the new field definition in the table and in each of its records. In each record, the modified fields are assigned default values that correspond to their new field definitions. In other words, the function replaces field definitions in each set of object data defined by the table and attached to an object.

The **fieldlist** argument has the same format as the **fieldlist** argument in [ade\\_odaddfield](#). The entry for this function has source code examples.

If a new field definition changes the field type, field values in existing records are converted to the new type if possible. This conversion may alter the values. For example, if you change the field type from **real** to **integer**, existing field values are converted by truncating their decimal parts.

The following sample creates a **resbuf** containing the field definitions to modify. **ade\_odmodifyfield()** uses this **resbuf** and returns a result code indicating a successful operation. Then it releases the **resbuf**, as required.

```
struct resbuf* pModifyFieldRb = NULL;
char* pszOdTable = "table1";
pModifyFieldRb = acutBuildList(
    RTSTR, "columns",
    RTLB,
    RTLB,
    RTSTR, "colname", RTSTR, "FIELD1",
    RTDOTE,
    RTLB,
    RTSTR, "coldesc", RTSTR, "Field 1 Description",
    RTDOTE,
    RTLB,
    RTSTR, "coltype", RTSTR, "integer",
    RTDOTE,
    RTLB,
    RTSTR, "defaultval", RTSTR, "0",
    RTDOTE,
    RTLE,
    // Define more fields as needed
    0);
int nResultCode = ade_odmodifyfield(pszOdTable, pModifyFieldRb);
if (RTNORM == nResultCode) {
    acutPrintf(
        "\nThe specified fields have been successfully modified.");
}
else {
    acutPrintf(
```

```
        "\nThe specified fields have not been modified.");  
    }  
    acutRelRb(pModifyFieldRb);
```



## ade\_odmodifytab

### [Object Data Functions](#)

---

Redefines a table.

```
int  
ade_odmodifytab(  
    struct resbuf* tab_defn);
```

Returns **RTNORM** or an error code.

**tab\_defn** List of table elements: the name of the table you will redefine, a new table description, and a sequence of new field definitions.

The **tab\_defn** argument has the same format as the **tab\_defn** argument in [ade\\_oddefinetab](#). The entry for this function has source code examples.

For the table you specify in the **tab\_defn** argument, the function replaces the existing table definition with the new one. For every object to which the table is attached, the corresponding fields of each record of the table are replaced. The old fields are deleted, and the new fields are assigned default values in accord with their field definitions.

The **ade\_odmodifyfield** function affects all active drawings in the drawing set. There should not be any queried objects for this operation.

**Note** This function will not operate unless your end user has superuser privileges.

The following sample creates a **resbuf** containing the updated table definition. **ade\_odmodifytab()** uses this **resbuf** and returns a result code indicating a successful operation. Then it releases the **resbuf**, as required.

```
struct resbuf* pOdTableModRb = acutBuildList(  
    RTLB,  
    RTSTR, "tablename", RTSTR, "table1",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "tabledesc", RTSTR, "New Sample Table",  
    RTDOT, RTDOT,  
    RTLB,  
    RTSTR, "columns",
```

```
    RTLB,  
    RTLB,  
        RTSTR, "colname", RTSTR, "field1",  
    RTDOTE,  
    RTLB,  
        RTSTR, "coldesc", RTSTR, "Field 1 column Description",  
    RTDOTE,  
    RTLB,  
        RTSTR, "coltype", RTSTR, "character",  
    RTDOTE,  
    RTLB,  
        RTSTR, "defaultval", RTSTR, "Default Value",  
    RTDOTE,  
    RTLE,  
    RTLE,  
    0);  
int nResultCode = ade_odmodifytab (pOdTableModRb);  
if (RTNORM == nResultCode) {  
    acutPrintf(  
        "\nThe specified table has been successfully modified.");  
}  
else {  
    acutPrintf(  
        "\nThe specified table has not been modified.");  
}  
acutRelRb(pOdTableModRb);
```



## ade\_odnewrecord

[Object Data Functions](#)

---

Defines a new object data record.

```
ade_id  
ade_odnewrecord(  
    char* table);
```

Returns a new record ID or **NULL**.

**table**     An existing table name to which the new record will belong.

The function creates a new record, populates its fields with default values according to the table definition, and returns the new record ID.



## ade\_odpresetfield

### [Object Data Functions](#)

---

Assigns a value to a field in a new record.

```
int
ade_odpresetfield(
    ade_id rec_id,
    char* field,
    struct resbuf* value);
```

Returns **RTNORM** or an error code.

**rec\_id** Record ID returned by **ade\_odnewrecord**.

**field** Field name.

**value** Field value.

Sets the value of a field in an Object Data record defined through **ade\_odnewrecord**.

The following sample creates a new object data record for "table1" using **ade\_odnewrecord()**. A **resbuf** is created which contains the new value to be applied to "field1" in the new object data record.

**Ade\_odpresetfield()** is called with all required parameters. A selection set containing all entities in the current drawing is created, then each entity has the new object data record attached using **ade\_odattachrecord()**. **Ade\_odfreerec()** is called to free the memory claimed in defining a new record and the **resbuf** is released as required.

```
char* pszOdTable = "table1";
ade_id recordId = ade_odnewrecord(pszOdTable);
struct resbuf* pOdFieldValRb = acutBuildList(RTSTR, "Newvalue",0);
char* pszOdField = "field1";
int returnCode = ade_odpresetfield(recordId, pszOdField, pOdFieldValRb);
ads_name selectionSet;
acedSSGet("_x", NULL, NULL, NULL, selectionSet);
ads_name ename;
long ssLength;
acedSSLength(selectionSet, &ssLength);
```

```
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        int returnCode = ade_odattachrecord(ename, recordId);
    }
}
returnCode = ade_odfreerec(recordId);
acutRelRb(pOdFieldValRb);
```



## ade\_odrecordqty

[Object Data Functions](#)

---

Counts the records attached to an object.

```
int  
ade_odrecordqty(  
    ads_name ename,  
    char* table);
```

Returns a record count or an error code.

**ename**      AutoCAD object name.

**table**      Table name can be up to 25 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.

The function counts how many records of the same table are attached to the object. See [ade\\_odaddrecord](#) for more information about attaching records to objects.



## ade\_odsetfield

### [Object Data Functions](#)

---

Sets a field value.

```
int
ade_odsetfield(
    ads_name ename,
    char* table,
    char* field,
    int recnum,
    struct resbuf* value);
```

Returns **RTNORM** or an error code.

<b>ename</b>	AutoCAD object name.
<b>table</b>	Table name can be up to 25 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>field</b>	Field name can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>recnum</b>	Record number; the first record number is <b>0</b> .
<b>value</b>	New field value.

To identify a unique record, you need to specify the table to which it belongs, the object to which it is attached, and its record number. The record number is necessary because more than one record from the same table can be attached to an object. For more information about records and record numbers, see [ade\\_odaddrecord](#).

The following sample creates a **resbuf** containing the updated field value. A selection set of all entities in the current drawing is created using **acedSSGet()**. Each entity in the selection set is passed to **ade\_odsetfield()** with all required parameters, the returned result code is checked for **RTNORM** and displayed. Then it releases the **resbuf**, as required.

```
char* pszOdTable = "table1";
ade_id recordId = ade_odnewrecord(pszOdTable);
```

```

struct resbuf* pOdSetFieldValRb = acutBuildList(RTSTR, "Newvalue", 0);
int recnum = 0;
char* pszOdField = "field1";
ads_name selectionSet;
ads_name ename;
acedSSGet("_x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        int nResultCode = ade_odsetfield(
            ename,
            pszOdTable,
            pszOdField,
            recnum,
            pOdSetFieldValRb);
        if (RTNORM == nResultCode) {
            acutPrintf(
                "\nThe field value for the specified record has been modified.");
        }
        else {
            acutPrintf(
                "\nThe field value for the specified record has not been modified.");
        }
    }
}
acutRelRb(pOdSetFieldValRb);

```



## ade\_odtabledefn

### [Object Data Functions](#)

---

Gets a table definition.

```
struct resbuf*  
ade_odtabledefn(  
    char* table);
```

Returns a table definition or **NULL**.

**table** Table name can be up to 25 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.

You must release the **resbuf**.

The table definition returned by **ade\_odtabledefn** has the same format as the **tab\_defn** argument in [ade\\_oddefinetab](#). The entry for this function has source code examples.

The following sample populates a **resbuf** with object data table schema information using **ade\_odtabledefn()**. The contents of the **resbuf** are displayed and the **resbuf** is released as required.

```
char* pszOdTable = "rirds";  
struct resbuf* pOdTableDefRb = ade_odtabledefn(pszOdTable);  
if(NULL != pOdTableDefRb) {  
    acutPrintf(  
        "\n\nThe following information was obtained about the \"%s\" table:"  
        , pszOdTable);  
    struct resbuf* rb = pOdTableDefRb;  
    while(NULL != rb) {  
        if (rb->restype == RTSTR) {  
            acutPrintf(  
                "\n\t\t\"%s\""  
                , rb->resval.rstring);  
        }  
        rb = rb->rbnext;  
    }  
}
```

```
acutRelRb(pOdTableDefRb);
```



## ade\_odtablelist

### [Object Data Functions](#)

---

Lists the tables in the drawing set.

```
struct resbuf*  
ade_odtablelist();
```

Returns a list of table names ([string](#)) or [NULL](#).

You must release the [resbuf](#).

The following sample parses the [resbuf](#) returned by [ade\\_odtablelist\(\)](#) and prints a list of object data tables in the current drawing. Then it releases the [resbuf](#), as required.

```
struct resbuf* pOdTableListRb = NULL;  
pOdTableListRb = ade_odtablelist();  
if (NULL != pOdTableListRb) {  
    acutPrintf(  
        "\nThe current drawing contains the following Object Data tables:");  
    int nOdTables = 0;  
    struct resbuf* rb = pOdTableListRb;  
    while(rb != NULL) {  
        ++nOdTables;  
        acutPrintf(  
            "\nObject data table %d is named: %s"  
            ,nOdTables, rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no object data tables in this project.");  
}  
acutRelRb(pOdTableListRb);
```



## ade\_osfexpand

[Other Functions](#)

---

Searches a directory and returns a list of file names.

```
struct resbuf*  
ade_osfexpand(  
    char* path,  
    char* extension,  
    char* pattern);
```

Returns a list of file names or **NULL**.

- path** Directory in which to search or **NULL**.  
If **NULL**, the function searches the working directory.
- extension** File name extension or **NULL**. If **NULL**, the function uses "dwg".
- pattern** Wild card pattern or **NULL**. If **NULL**, the function uses "\*" (search for all file names with the given extension and path).

You must release the **resbuf**.

For information about wild card patterns, look up "wild-card characters" on the Index tab of AutoCAD Map Help.

The following sample parses the **resbuf** returned by **ade\_osfexpand()** and prints a list of matching files. Then it releases the **resbuf**, as required.

```
char* pszFilePath = "C:\\MyFiles";  
char* pszFileExtension = "dwg";  
char* pszFilePattern = NULL;  
struct resbuf* pFileNamesRb = ade_osfexpand(pszFilePath, pszFileExtension, pszFilePattern);  
while(pFileNamesRb != NULL) {  
    acutPrintf(  
        "\n\nThe following file of type \"%s\" were found in %s: %s"  
        , pszFileExtension, pszFilePath, pFileNamesRb->resval.rstring);  
    pFileNamesRb = pFileNamesRb->rbnext;  
}
```

```
acutRelRb(pFileNamesRb);
```



## ade\_prefgetval

### [Option Functions](#)

Gets an AutoCAD Map option setting.

```
struct resbuf*  
ade_prefgetval(  
    char* variable);
```

Returns an option value or **NULL**.

**variable**     Option name. See the Options tables below.

You must release the **resbuf**.

The function return value depends on which option you specify. The tables below show option names and return values, organized by option type.

### Work Session Options

RestoreLastActiveDwgsOnStartup	-1 or NULL.
ActivateDwgsOnAttach	-1 or NULL.
DontAddObjectsToSaveSet	-1 or NULL.
MarkObjectsForEditingWithoutPrompting	-1 or NULL.
LogFileActive	-1 or NULL.
LogFileName	File name ( <b>RTSTR</b> ). For example, "ade.log".
LogMessageLevel	0, 1, or 2.

### Query Options

QueryFileDirectory.	Path ( <b>RTSTR</b> ). For example, "c:\\data\\qry".
CaseSensitiveMatch.	-1 or NULL.
SaveCurrQueryInSession.	-1 or NULL.

MkSelSetWithQryObj	-1 or NULL.
DefaultJoinOperator	1 = OR, 2 = AND.
ColorForAdd	<a href="#">Color</a> (RTSTR).
ColorForRemove	<a href="#">Color</a> (RTSTR).
BlockLocnForQuery	1 = insertion point, 2 = bounding box.
TextLocnForQuery	1 = insertion point, 2 = bounding box.
ShowBlockAsInsPt	-1 or NULL.
ShowImageAsBoundary	-1 or NULL.
CreateAssociativeHatchObjects	-1 or NULL.
ReferenceBoundaryForAreaLocation	-1 or NULL.

### Save Back Options

RedefineBlockDefinitions	-1 or NULL.
RedefineLayerDefinitions	-1 or NULL.
RedefineTextStyleDefinitions	-1 or NULL.
RemoveUnusedGroups	-1 or NULL.
EraseSavedBackObjects	-1 or NULL.
RemoveLockAfterSave	-1 or NULL.
CreateHistoryFileOfChanges	-1 or NULL.
CreateBackupFileOfSourceDwg	-1 or NULL.

### External Database Options

NoOfSQLConditionsInHistory	RTSHORT.
DisplayTabsInSingleView	-1 or NULL.
OpenDataViewReadOnly	-1 or NULL.
SaveDataViewFmtChanges	-1 or NULL.
ReconnectDbOnWSOpen	-1 or NULL.

ShowFullDBPath	-1 or NULL.
KeepDataViewOnTop	-1 or NULL.
dbfDatabases	RTSTR, one of the following: "Prompt", "DB3", "DB4", "DB5", "FOX2.0", "FOX2.5", or "FOX2.6".
xlsDatabases	RTSTR, one of the following: "Prompt", "Excel3", "Excel4", "Excel5", or "Excel7".
dbDatabases	RTSTR, one of the following: "Prompt", "Paradox3.0", "Paradox4.0", or "Paradox5.0".

### Coordinate Transformation Options

AdjustSizesAndScalesForChangesInUnits	-1 or NULL.
AdjustRotationsForMapDistortions	-1 or NULL.
AdjustSizesAndScalesForMapDistortions	-1 or NULL.
AdjustElevations	-1 or NULL.
AdjustZeroRotationObjects	-1 or NULL.

### System Options

AccessWorkCenter	-1 or NULL.
CheckoutDirectory	Path (RTSTR). For example, "c:\\data\\dwg" or "" if none.
PreserveAWCFiles	-1 or NULL.
ForceUserLogin	-1 or NULL.
EnableObjectLocking	-1 or NULL.
ReadPrefFromINI	-1 or NULL.
NumberOfOpenDwgs	RTSHORT
DoublePrec	RTREAL, 0 or greater, but less than 1.

The "ForceUserLogin" and "DoublePrec" system options cannot be modified unless your end user has

superuser privileges.

If "DoublePrec" is set to 0, the behavior of data extension queries is the same as before introducing this option. The "DoublePrec" option has no user interface equivalent.

### Workspace Options

CheckClasses	-1 or NULL.
CheckDrawings	-1 or NULL.
CheckQueryLibrary	-1 or NULL.
CheckDatabases	-1 or NULL.
CheckTables	-1 or NULL.
CheckQueries	-1 or NULL.
CheckTopologies	-1 or NULL.
CheckLPNs	-1 or NULL. Note that link path names (LPNs) have been replaced by link templates in AutoCAD Map.
ShowOPMOnStartup	-1 or NULL.
ShowWSpaceOnStartup	-1 or NULL.
WSpaceDockingView	-1 or NULL.
WSpaceWindowRect	A list of four values (RTSHORT) that define the left, top, right, and bottom of the window rectangle.

Database tables and database query categories are visible in the workspace only if "CheckTables" and "CheckQueries" are set to T and "CheckDatabases" is set to T also.

The following workspace options are read only. That is, they can be used only with [ade\\_prefgetval](#) to determine if a category is visible in the workspace.

### Read-Only Workspace Options

ClassesVisible	-1 or NULL.
DrawingsVisible	-1 or NULL.
QueryLibraryVisible	-1 or NULL.
DatabasesVisible	-1 or NULL.

TablesVisible	-1 or NULL.
QueriesVisible	-1 or NULL.
TopologiesVisible	-1 or NULL.
LPNsVisible	-1 or NULL.

The following sample parses the **resbuf** returned by **ade\_prefgetval()** and prints the value for the specified option using a **switch case statement**. Then it releases the **resbuf**, as required.

```

char* pszOptionVar = "LogFileName";
struct resbuf* pGetPrefValRb = ade_prefgetval(pszOptionVar);
if(pGetPrefValRb != NULL){
    struct resbuf* rb = pGetPrefValRb;
    while(rb != NULL) {
        switch(rb->restype)
        {
            case RTT:
                acutPrintf(
                    "\nThe %s option variable contained the value: %d"
                    , pszOptionVar, rb->resval.rint);
                break;
            case RTSHORT:
                acutPrintf(
                    "\nThe %s option variable contained the value: %d"
                    , pszOptionVar, rb->resval.rint);
                break;
            case RTSTR:
                acutPrintf(
                    "\nThe %s option variable contained the value: %s"
                    , pszOptionVar, rb->resval.rstring);
                break;
            default:
                acutPrintf(
                    "\nCould not determine the value");
                break;
        }
        rb = rb->rbnext;
    }
}
else
{

```

```
    acutPrintf("\nNo value was assigned to %s."
              , pszOptionVar);
}
acutRelRb(pGetPrefValRb);
```



## ade\_prefsetval

### [Option Functions](#)

---

Sets an AutoCAD Map option.

```
int  
ade_prefsetval(  
    char* variable,  
    struct resbuf* value);
```

Returns **RTNORM** or an error code.

**variable**     Option name.  
**value**        Value appropriate for the given option (type varies).

See [ade\\_prefgetval](#) for a list of option names and values.

The following example sets "ColorForAdd" to "red".

```
ade_prefsetval ("ColorForAdd", rb);
```

The argument **rb** points to a **resbuf** structure containing

```
rb->resval.rstring = "red"
```

You can create a new **resbuf** structure with [ads\\_newrb\(\)](#).

The following sample creates a **resbuf** containing the updated option value. [ade\\_prefsetval\(\)](#) is called with all required parameters, the returned result code is checked for **RTNORM** and displayed. Then it releases the **resbuf**, as required.

```
char* pszOptionVar = "LogFileName";  
struct resbuf* pSetPrefValRb = acutBuildList(RTSTR, "ade.log", 0);  
int nResultCode = ade_prefsetval(pszOptionVar, pSetPrefValRb);  
if (RTNORM == nResultCode) {  
    acutPrintf(  
        "\nThe %s option variable has been successfully modified to: %s"  
        , pszOptionVar, pSetPrefValRb->resval.rstring);  
}
```

```
}  
else {  
    acutPrintf(  
        "\nThe specified options variable has not been modified.");  
}  
acutRelRb(pSetPrefValRb);
```



## ade\_projgetctgname

[Coordinate Transformation Functions](#)

---

Identifies the category that a coordinate system belongs to.

char\*

```
ade_projgetctgname(  
    char* cscode);
```

Returns a coordinate system category or **NULL**.

**cscode**      Coordinate system code, eight characters.



## ade\_projgetinfo

### [Coordinate Transformation Functions](#)

---

Gets information about a projection system.

```
struct resbuf*
ade_projgetinfo(
    char* cscode,
    char* info_type);
```

Returns a piece of coordinate system information or **NULL**.

**cscode**        Coordinate system code, eight characters.  
**info\_type**    Information type; see Information Types below.

You must release the **resbuf**.

### Information Types

<b>description</b>	Description. For example, "World Geodetic System of 1984 Latitude/Longitude in Degrees".
<b>projection</b>	Projection. For example, "Unity Conversion, produce/accept lat/longs".
<b>datum</b>	Datum. For example, "North American Datum of 1927, Mean Values".

The following sample parses the **resbuf** returned by **ade\_projgetinfo()** and prints the value for the specified option. Then it releases the **resbuf**, as required.

```
char* pszCategoryCode = "MO-W";
char* pszInfoType = "description";
pCoordSysRb = ade_projgetinfo(pszCategoryCode, pszInfoType);
struct resbuf* rb = pCoordSysRb;
while(pCoordSysRb != NULL) {
    acutPrintf(
        "\nThe following %s information is available for the %s coordinate system:\n\n\t%s"
        , pszInfoType, pszCategoryCode, pCoordSysRb->resval.rstring);
```

```
pCoordSysRb = pCoordSysRb->rbnext;  
}  
acutRelRb(pCoordSysRb);
```



## ade\_projgetwscode

[Coordinate Transformation Functions](#)

---

Gets the current drawing's coordinate system code.

```
char*  
ade_projgetwscode();
```

Returns a coordinate system code or the empty string.



## ade\_projlistcrdsysts

### [Coordinate Transformation Functions](#)

---

Lists available coordinate systems in a given category.

```
struct resbuf*  
ade_projlistcrdsysts(  
    char* categoryName);
```

Returns a list of available coordinate systems or **NULL**.

**categoryname**      Coordinate system code, eight characters.

The following sample parses the **resbuf** returned by **ade\_projlistcrdsysts()** and prints the value for the specified option. Then it releases the **resbuf**, as required.

```
char* pszCategoryName = "Lat Longs";  
struct resbuf* pCoordSysRb = ade_projlistcrdsysts(pszCategoryName);  
if(pCoordSysRb != NULL){  
    acutPrintf(  
        "\nThe following coordinate system(s) have been detected for the %s category:"  
        , pszCategoryName);  
    struct resbuf* rb = pCoordSysRb;  
    while(rb != NULL) {  
        acutPrintf(  
            "\n\n\t %s"  
            , rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else  
{  
    acutPrintf(  
        "\nNo coordinate system(s) have been detected for the %s category: %s"  
        , pszCategoryName);  
}  
acutRelRb(pCoordSysRb);
```





## ade\_projlistctgy

### [Coordinate Transformation Functions](#)

---

Lists available coordinate system categories.

```
struct resbuf*  
ade_projlistctgy();
```

Returns a list of coordinate system categories or **NULL**.

The following sample parses the **resbuf** returned by **ade\_projlistctgy()** and prints the coordinate system category list. Then it releases the **resbuf**, as required.

```
struct resbuf* pCoordSysCtgyRb = ade_projlistctgy();  
if(pCoordSysCtgyRb != NULL) {  
    acutPrintf(  
        "\n\nThe following coordinate system categories have been detected:");  
    struct resbuf* rb = pCoordSysCtgyRb;  
    while(rb != NULL) {  
        acutPrintf(  
            "\n\n\t%s"  
            , rb->resval.rstring);  
        rb = rb->rbnext;  
    }  
}  
else  
{  
    acutPrintf(  
        "\n\nNo coordinate system categories have been detected");  
}  
acutRelRb(pCoordSysCtgyRb);
```



## ade\_projptbackward

[Coordinate Transformation Functions](#)

---

Converts point coordinates from destination coordinate system to source.

```
int  
ade_projptbackward(  
    ads_point pt,  
    ads_point result);
```

Returns **RTNORM** or an error code.

**pt** Destination point to convert, a set of 2D or 3D coordinate values. If 3D, the Z value is ignored.

**result** Corresponding source values.

Before you can use **ade\_projptbackward** to convert points, you must first identify the coordinate systems that you are converting between. Use [ade\\_projsetsrc](#) to set the source system and [ade\\_projsetdest](#) to set the destination system. The **ade\_projptbackward** function assumes that the coordinate values you pass to it belong to the destination system, and it returns corresponding source values. The [ade\\_projptforward](#) function does the inverse.

The following example prompts the user for a destination point and returns the corresponding source coordinates through the **result** parameter.

```
ads_point pt1;  
ads_point result;  
ads_getpoint (NULL, "Pick a point", pt1);  
ade_projptbackward (pt1, result);
```

For more information, see [Converting Coordinates](#).



## ade\_projptforward

### [Coordinate Transformation Functions](#)

---

Converts point coordinates from source coordinate system to destination.

```
int  
ade_projptforward(  
    ads_point pt,  
    ads_point result);
```

Returns **RTNORM** or an error code.

**pt**            Source point to convert, a set of 2D or 3D coordinate values. If 3D, the Z value is ignored.

**result**        Corresponding destination values.

Before you can use **ade\_projptforward** to convert points, you must first identify the coordinate systems that you are converting between. Use [ade\\_projsetsrc](#) to set the source system and [ade\\_projsetdest](#) to set the destination system. The **ade\_projptforward** function assumes that the coordinate values you pass to it belong to the source system, and it returns corresponding destination values. The [ade\\_projptbackward](#) function does the inverse.

The following example prompts the user for a source point and passes the corresponding destination coordinates through the **result** parameter.

```
ads_point pt1;  
ads_point result;  
ads_getpoint (NULL, "Pick a point", pt1);  
ade_projptforward (pt1, result);
```

For more information, see [Converting Coordinates](#).



## ade\_projsetdest

### [Coordinate Transformation Functions](#)

---

Sets the destination coordinate system for converting points.

```
int  
ade_projsetdest(  
    char* cscode);
```

Returns **RTNORM** or an error code.

***cscode***     Coordinate system code, eight characters.

Before you can use either [ade\\_projptforward](#) or [ade\\_projptbackward](#) to convert points, you must first identify the coordinate systems that you are converting between. Use **ade\_projsetdest** to set the destination system and [ade\\_projsetsrc](#) to set the source system. The **ade\_projptforward** function assumes that the coordinate values you pass to it belong to the source system, and it returns corresponding destination values. The **ade\_projptbackward** function does the inverse.

For more information, see [Converting Coordinates](#).



## ade\_projsetsrc

### [Coordinate Transformation Functions](#)

---

Sets the source coordinate system for converting points.

```
int  
ade_projsetsrc(  
    char* cscode);
```

Returns **RTNORM** or an error code.

**cscode**      Coordinate system code, eight characters.

Before you can use either [ade\\_projptforward](#) or [ade\\_projptbackward](#) to convert points, you must first identify the coordinate systems that you are converting between. Use **ade\_projsetsrc** to set the source system and [ade\\_projsetdest](#) to set the destination system. The **ade\_projptforward** function assumes that the coordinate values you pass to it belong to the source system, and it returns corresponding destination values. The **ade\_projptbackward** function does the inverse.

For more information, see [Converting Coordinates](#).



## ade\_projsetwscode

[Coordinate Transformation Functions](#)

---

Sets the coordinate system for the current drawing.

```
int  
ade_projsetwscode(  
    char* cscode);
```

Returns **RTNORM** or an error code.

**cscode**      Coordinate system code, eight characters.



## ade\_qldelctgy

[Query Library Functions](#)

---

Deletes a query library category.

int

```
ade_qldelctgy(  
    ade_id ctgy_id);
```

Returns **RTNORM** or an error code.

**ctgy\_id**      Category ID.



## ade\_qldelquery

[Query Library Functions](#)

---

Deletes a query from the query library.

int

```
ade_qldelquery(  
    ade_id qry_id);
```

Returns **RTNORM** or an error code.

**qry\_id**      Query ID.



## ade\_qlgetctgyinfo

### [Query Library Functions](#)

---

Gets information about a query category.

```
struct resbuf*
ade_qlgetctgyinfo(
    ade_id ctg_id,
    char* info);
```

Returns the requested information or **NULL**.

**ctgy\_id**      Category ID.

**info**          Type of category information to get: **"name"** to get the category name, or **"qrylist"** to get a list of query IDs of the queries in the category.

You must release the **resbuf**.

The information returned depends on the **info** argument you use, but it is always in list format. For example:

The following sample populates a **resbuf** with query category id's using **ade\_qllistctgy()**. This **resbuf** is then used by **ade\_qlgetctgyinfo()** to populate a **resbuf** containing category names or category id's depending on the value of **info**. The switch case statement prints the appropriate information, (category names or query Id's). Then it releases the **resbuf**, as required.

```
struct resbuf* pQryCtgyRb = NULL;
char* pszCategoryInfo = "qrylist";
struct resbuf* pQryCtgyId = ade_qllistctgy();
if(pQryCtgyId != NULL) {
    struct resbuf* rb = pQryCtgyId;
    while(rb != NULL) {
        pQryCtgyRb = ade_qlgetctgyinfo(rb->resval.rreal, pszCategoryInfo);
        if(pQryCtgyRb != NULL) {
            switch(pQryCtgyRb->restype)
            {
                case RTREAL:
                    acutPrintf(
                        "\nThe %s option you've specified produced the following query category id: %.0lf"
```

```
        , pszCategoryInfo, pQryCtgyRb->resval.rreal);
    break;
case RTSTR:
    acutPrintf(
        "\nThe %s option you've specified produced the following query category name: %s"
        , pszCategoryInfo, pQryCtgyRb->resval.rstring);
    break;
default:
    acutPrintf(
        "\nCould not determine the value");
    break;
    }
}
rb = rb->rbnext;
}
}
else
{
    acutPrintf(
        "\nThe %s category you've requested produced no information"
        , pszCategoryInfo);
}
acutRelRb(pQryCtgyId);
acutRelRb(pQryCtgyRb);
```



## ade\_qlgetqryinfo

### [Query Library Functions](#)

---

Gets information about a query.

```
struct resbuf*  
ade_qlgetqryinfo(  
    ade_id qry_id,  
    char* info);
```

Returns the requested information or **NULL**.

**qry\_id**      Query ID.

**info**        Information type. See the Information Types table below.

You must release the **resbuf**.

### Information Types

<b>name</b>	Query name.
<b>description</b>	Query description.
<b>category</b>	Category name.
<b>qtype</b>	Query type: <b>1</b> = internal, <b>2</b> = external.
<b>filename</b>	For an externally saved query, full path name ( <b>string</b> ).

A query gets a name and an ID when it is saved to a query category of the query library. A new query that you have not yet saved does not have a name or an ID.

The following sample obtains a query ID using **ade\_qlqrygetid()**. **Ade\_qlgetqryinfo()** is called with all required parameters and status information is displayed based on the **resbuf** returned by the function. The **resbuf** is then released as required.

```
char* pszQueryName = "LocAll";  
ade_id queryId = ade_qlqrygetid(pszQueryName);
```

```
char* pszQueryInfo = "category";
struct resbuf* pQueryInfoRb = ade_qlgetqryinfo(
    queryId,
    pszQueryInfo);
if (NULL != pQueryInfoRb) {
    acutPrintf(
        "\nThe \"%s\" query is contained within the \"%s\" query category."
        ,pszQueryName, pQueryInfoRb->resval.rstring);
}
else {
    acutPrintf(
        "\nThe requested information could not be found.");
}
acutRelRb(pQueryInfoRb);
```



## ade\_qllistctgy

### [Query Library Functions](#)

---

Lists the query category IDs.

```
struct resbuf*  
ade_qllistctgy();
```

Returns a list of category IDs, or, if there are no categories or an error occurs, **NULL**.

You must release the **resbuf**.

The following sample parses the **resbuf** returned by **ade\_qllistctgy()** and prints a list of query category ID's. Then it releases the **resbuf**, as required.

```
struct resbuf* pQueryCategoryListRb = NULL;  
pQueryCategoryListRb = ade_qllistctgy();  
if (NULL != pQueryCategoryListRb) {  
    int nQueryCategories = 0;  
    struct resbuf* rb = pQueryCategoryListRb;  
    while(rb != NULL) {  
        ++nQueryCategories;  
        acutPrintf(  
            "\nQuery category %d has the Id: %.0lf",  
            nQueryCategories, rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no query categories in this project.");  
}  
acutRelRb(pQueryCategoryListRb);
```



## ade\_qlloadqry

[Query Library Functions](#)

---

Makes a saved query current.

```
int  
ade_qlloadqry(  
    ade_id qry_id);
```

Returns **RTNORM** or an error code.

**qry\_id**      Query ID.

Once loaded, the query becomes the current query. If there is already a current query, this query replaces it.



## ade\_qlqrygetid

[Query Library Functions](#)

---

Gets a query ID.

```
ade_id  
ade_qlqrygetid  
    char* query_name);
```

Returns a query ID or **ADE\_NULLID**.

**query\_name**      Query name.

A query gets a name and an ID when it is saved to a query category of the query library. A new query that you have not yet saved does not have a name or an ID.

The query name is enough to identify a query uniquely. The category name is not required. Within the current drawing and all attached source drawings, no two queries can have the same name, even if they are saved in different categories.



## ade\_qlsetctgname

[Query Library Functions](#)

---

Changes a query category name.

```
int  
ade_qlsetctgname(  
    ade_id ctgy_id,  
    char* name);
```

Returns **RTNORM** or an error code.

<b>ctgy_id</b>	Query category ID.
<b>name</b>	Name of new category, up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character

The category name cannot contain spaces.



## ade\_qlsetquery

### [Query Library Functions](#)

---

Changes a query name, description, or the category it belongs to.

```
int  
ade_qlsetquery(  
    ade_id qry_id,  
    char* info,  
    struct resbuf* value);
```

Returns **RTNORM** or an error code.

**qry\_id**      Query ID.  
**info**        Type of information to modify. See the Query Properties table below.  
**value**        New value (type varies).

### Query Properties

<b>name</b>	Name of query ( <b>RTSTR</b> ), up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>description</b>	Description of query ( <b>RTSTR</b> ), up to 132 characters long. Can contain spaces. Must be unique and start with an alphanumeric character.
<b>category</b>	Category ID, ( <b>RTREAL</b> ).

This function does not change file name or storage type.

A query gets a name and an ID when it is saved to a query category of the query library. A new query that you have not yet saved does not have a name or an ID.

The following sample creates a **resbuf** containing the updated option value. The id for the query to be modified is then obtained with **ade\_qlqrygetid()**. **Ade\_qlsetquery()** is called with all required parameters, the returned result code is checked for **RTNORM** and displayed. Then it releases the **resbuf**, as required.

```
char* pszQueryProperty = "name";  
char* pszQueryName = "RtList";
```

```
struct resbuf* pQueryPropValRb = acutBuildList(RTSTR, "MyRenamedQuery", 0);
ade_id queryId = ade_qlqrygetid(pszQueryName);
if(queryId != ADE_NULLID) {
    int returnCode = ade_qlsetquery(queryId, pszQueryProperty, pQueryPropValRb);
    if (RTNORM == returnCode) {
        acutPrintf(
            "\nThe query %s property has been modified."
            , pszQueryProperty);
    }
    else {
        acutPrintf(
            "\nThe query %s property has not been modified."
            , pszQueryProperty);
    }
}
acutRelRb(pQueryPropValRb);
```



## ade\_qryclear

[Query Functions](#)

---

Clears the current query.

```
int  
ade_qryclear();
```

Returns **RTNORM** or an error code.



## ade\_qrydefine

### [Query Functions](#)

---

Defines a query.

```
ade_id  
ade_qrydefine(  
    char* joinop,  
    char* bgroups,  
    char* not_op,  
    char* condtype,  
    struct resbuf* qrycond,  
    char* endgroups);
```

Returns the ID of the query condition it creates or **ADE\_NULLID**.

<b>joinop</b>	A joining operator: "and" or "or" or "" (none). If "" (none) is specified, the default joining operator is used (see <a href="#">ade_prefgetval</a> ).
<b>bgroups</b>	For grouping this condition with others in the query definition you are building. Use one or more open parentheses as needed, or "" (none). For example, "(".
<b>not_op</b>	The NOT operator, if needed: "not" or "" (none).
<b>condtype</b>	A condition type: "Location", "Property", "Data", or "SQL".
<b>qrycond</b>	A condition expression. Depends on the condition type. See <a href="#">Condition Expressions</a> below.
<b>endgroups</b>	For grouping this condition with others in the query definition you are building. Use one or more close parentheses as needed, or "" (none). For example, ")".

A query definition is composed of one or more conditions, each defined by a separate **ade\_qrydefine** call. You can group conditions by supplying parentheses or empty strings to the **bgroups** or **endgroups** parameters as needed.

You must specify all six **ade\_qrydefine** arguments.

### Condition Expressions

The **qrycond** parameter requires a condition expression. Condition expressions are lists. What you include

in the list depends on the condition type: Location, Property, Data, or SQL.

[Location Expressions](#) [Property Expressions](#)

[Data Expressions](#)

[SQL Expressions](#)

You must release the **resbuf**.

The following sample creates a **resbuf** containing the query condition values, (objects classified as Ponds). **ade\_qrydefine()** is called with all required parameters, the returned **queryId** is checked for **ADE\_NULLID** and an appropriate message is displayed. Then it releases the **resbuf**, as required.

```
char* pszJoinOperator = ""; // none
char* pszBgnCondGrouping = ""; // none
char* pszNotOperator = ""; // none
char* pszCondType = "Property";
char* pszEndCondGrouping = ""; // none
struct resbuf* pQueryConditionRb = acutBuildList(
    RTLB,
    RTSTR, "feature",
    RTSTR, "=",
    RTSTR, "Pond",
    RTLE,
    0 );
ade_id queryId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pQueryConditionRb,
    pszEndCondGrouping);

if (queryId != ADE_NULLID) {
    acutPrintf(
        "\nA %s query has been defined."
        , pszCondType);
}
else {
    acutPrintf(
        "\nA %s query was not defined."
        , pszCondType);
}
acutRelRb(pQueryConditionRb);
```



## ade\_qryexecute

### [Query Functions](#)

---

Executes the current query.

```
ads_real  
ade_qryexecute();
```

Returns the number of queried objects. If none, it returns **0.0**.

Executing a query makes a new selection set of the queried objects if the "**MkSelSetWithQryObj**" option is turned on, as in the following:

```
struct resbuf* pSetPrefValRb = acutBuildList(RTT, 0);  
ade_prefsetval("MkSelSetWithQryObj", pSetPrefValRb);  
acutRelRb(pSetPrefValRb);
```

The "**P**" argument identifies the "previous" selection set (the objects currently or most recently selected).

**Note** Whenever you create a selection set, you replace the previous selection set. Make sure you know which objects you are getting. The will query run slower in this case because of the extra work involved.

The following sample sets the **MkSelSetWithQryObj** option which will place the resulting queried objects in a selection set of your specification. **Ade\_qryexecute()** is called and the number of objects placed into the selection set is displayed. The **Resbuf** is released as required and the selection set is freed.

```
ads_name ssQueriedObjects;  
char* pszOptionVar = "MkSelSetWithQryObj";  
struct resbuf* pSetPrefValRb = acutBuildList(RTT, 0);  
int nResultCode = ade_prefsetval(pszOptionVar, pSetPrefValRb);  
if (RTNORM == nResultCode) {  
    ads_real queriedObjects = ade_qryexecute();  
    if (ADE_NULLID != queriedObjects) {  
        acedSSGet("P", NULL, NULL, NULL, ssQueriedObjects);  
        long ssQueriedObjectsLength;  
        acedSSLength(ssQueriedObjects, &ssQueriedObjectsLength);  
        acutPrintf(  
            "\n%d objects have been placed in the \"QueriedObjects\" selection set."  
            , ssQueriedObjectsLength);  
    }  
}
```

```
}
else {
    acutPrintf(
        "\nNo objects were queried.");
}
}
else {
    acutPrintf(
        "\nThe specified options variable has not been modified.");
}
acutRelRb(pSetPrefValRb);
int resultCode = acedSSFree(ssQueriedObjects);
```



## ade\_qrygetcond

### [Query Functions](#)

---

Gets a condition of the current query.

```
struct resbuf*  
ade_qrygetcond(  
    ade_id condition_id);
```

Returns a query condition or **NULL**.

**condition\_id**      Query condition ID.

You must release the **resbuf**.

See [ade\\_qrydefine](#) for information about query conditions.

The following sample shows the use of **ade\_qrygetcond()** only. A **resbuf** is created, the **resbuf** is filled based on the return of **ade\_qrygetcond()**. Then it releases the **resbuf**, as required.

```
struct resbuf* pNewQueryConditionRb = NULL;  
pNewQueryConditionRb = ade_qrygetcond(queryId);  
acutRelRb(pQueryConditionRb);
```

The following sample shows the use of **ade\_qrygetcond()** in conjunction with **ade\_qrydefine()**.

```
char* pszJoinOperator = ""; // none  
char* pszBgnCondGrouping = ""; // none  
char* pszNotOperator = ""; // none  
char* pszCondType = "Property";  
char* pszEndCondGrouping = ""; // none  
struct resbuf* pQueryConditionRb = acutBuildList(  
    RTLB,  
        RTSTR, "feature",  
        RTSTR, "=",  
        RTSTR, "Pond",  
    RTLE,  
    0 );
```

```

ade_id queryId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pQueryConditionRb,
    pszEndCondGrouping);
if (queryId != ADE_NULLID) {
    acutPrintf(
        "\nA query has been defined with the following condition:\n\n");
}
else {
    acutPrintf(
        "\nThere are no defined queries.");
}
acutRelRb(pQueryConditionRb);
struct resbuf* pNewQueryConditionRb = ade_qrygetcond(queryId);
if(NULL != pNewQueryConditionRb) {
    struct resbuf* rb = pNewQueryConditionRb;
    while(NULL != rb) {
        if(rb->restype == RTSTR){
            acutPrintf(
                "\n%s\n "
                , rb->resval.rstring);
        }
        rb = rb->rbnext;
    }
}
else{
    acutPrintf(
        "\nNo information could be retrieved.");
}
acutRelRb(pNewQueryConditionRb);

```

After running the code, take notice of how the command line display matches the `ade_qrydefine()` parameter list.



## ade\_qrygetdwgandhandle

### [Query Functions](#)

---

Gets the source drawing ID and original handle of a queried object.

```
struct resbuf*
ade_qrygetdwgandhandle(
    ads_name ename);
```

Returns the drawing ID and handle for the queried object or **NULL**.

**ename** AutoCAD entity name.

You must release the **resbuf**.

This function returns the ID of the source drawing from which the object was queried and the handle by which the object is known in that drawing.

The following sample creates a selection set of all entities in the current drawing using **acedSSGet()**. Each entity in the selection set is passed to **ade\_qrygetdwgandhandle()** with all required parameters, the results of which populate a **resbuf**. The **resbuf** values are displayed and then released, as required.

```
ads_name ename;
ads_name selectionSet;
struct resbuf* pQueriedObjInfoRb = NULL;
acedSSGet("_x", NULL, NULL, NULL, selectionSet);
long ssLength;
acedSSLength( selectionSet, &ssLength; );
for( int i = 0; i < ssLength; ++i )
{
    if( acedSSName(selectionSet, i, ename) == RTNORM )
    {
        pQueriedObjInfoRb = ade_qrygetdwgandhandle(ename);
        if(NULL != pQueriedObjInfoRb) {
            struct resbuf* rb = pQueriedObjInfoRb;
            while(NULL != rb) {
                switch(rb->restype)
                {
```

```
case RTREAL:
    acutPrintf(
        "\n\nThe queried objects drawing Id is: %.0lf"
        , rb->resval.rreal);
    break;
case RTSTR:
    acutPrintf(
        "\nThe queried objects entity handle is: %s"
        , rb->resval.rstring);
    break;
default:
    acutPrintf(
        "\nCould not determine the value");
    break;
}
rb = rb->rbnext;
}
}
}
}
acutRelRb(pQueriedObjInfoRb);
```



## ade\_qrygetentlist

### [Query Functions](#)

---

Returns the list of entity handles for all objects that satisfy the current query in a specific drawing.

```
struct resbuf*
ade_qrygetentlist(
    ade_id dwg_id);
```

Returns a list of the handles of selected objects or **NULL**.

**dwg\_id** Drawing ID of the drawing to query.

This function executes the current query and finds all objects that satisfy it in the drawing specified by the **dwg\_id** argument. The entity handles of the objects are returned to the calling function.

Once you have the handle to an object, you can get the entity name with the [ade\\_qryhandent](#) function and use it to perform other functions. For example, you could use [entget](#) (and [ads\\_entget](#)) to retrieve the entity and its definition data.

You must release the **resbuf**.

The following sample populates a **resbuf** with attached drawing id's using [ade\\_qllistctgy\(\)](#). This **resbuf** contains the input parameter used by [ade\\_qrygetentlist\(\)](#) to populate a **resbuf** containing queried entity id's. Those id's are displayed, and the **resbufs**, are released as required.

```
struct resbuf* pQueriedObjHandlesRb = NULL;
struct resbuf* pDsDwgIdRb = ade_dslist(ADE_NULLID, ADE_FALSE);
if (NULL != pDsDwgIdRb) {
    struct resbuf* rb = pDsDwgIdRb;
    while(NULL != rb) {
        struct resbuf* pQueriedObjHandlesRb = ade_qrygetentlist(rb->resval.rreal);
        if (NULL != pQueriedObjHandlesRb) {
            struct resbuf* rbHand = pQueriedObjHandlesRb;
            while(NULL != rbHand) {
                acutPrintf(
                    "\nThis queried objects entity handle is: %s"
                    , rbHand->resval.rstring);
                rbHand = rbHand->rbnext;
            }
        }
        rb = rb->rbnext;
    }
}
```

```
    }
    rb = rb->rbnext;
  }
  else{
    acutPrintf(
      "\nNo information could be retrieved.");
  }
}
}
else {
  acutPrintf(
    "\nNo queried objects were returned for this project.");
}
acutRelRb(pDsDwgIdRb);
acutRelRb(pQueriedObjHandlesRb);
```



## ade\_qrygetretransform

### [Query Functions](#)

---

Checks whether transformation is enabled for the current report query.

```
int  
ade_qrygetretransform();
```

Returns **TRUE** or **FALSE**.

The following sample enables coordinate transformation for a report query using `ade_qrysetretransform()`, then checks the status of the transformation flag using `ade_qrygetretransform()`.

```
ade_boolean bTxfmForReportQry = ADE_TRUE;  
int resultCode = ade_qrysetretransform(bTxfmForReportQry);  
if (RTNORM == resultCode){  
    acutPrintf(  
        "\nReport query transformation has been set.");  
}  
else {  
    acutPrintf(  
        "\nNo transformation flag has been set.");  
}  
int nIsEnabled = ade_qrygetretransform();  
if (1 == nIsEnabled){  
    acutPrintf(  
        "\nReport query transformation is enabled.");  
}  
else {  
    acutPrintf(  
        "\nReport query transformation is not enabled.");  
}
```



## ade\_qrygroup

### [Query Functions](#)

---

Groups a sequence of two or more query conditions.

```
int
ade_qrygroup(
    ade_id condition_id1,
    ade_id condition_id2);
```

Returns **RTNORM** or an error code.

**condition\_id1** ID of first condition of the group.  
**condition\_id2** ID of last condition of the group.

This function affects the current query.

A query definition consists of a sequence of query conditions. Within such a sequence, you can define subsequences of two or more conditions by grouping them (by enclosing them in parentheses). You can group conditions when you first define the query. See the **bggroups** and **endgroups** parameters of [ade\\_qrydefine](#). Or you can do it later using **ade\_qrygroup**.

When you call **ade\_qrygroup**, the condition you specify as the first condition of the group (**condition\_id1**) must be a predecessor to the one you specify as the last (**condition\_id2**). The function groups the first and the last and any conditions in between.

The following sample defines a location condition which utilizes a buffer fence selection option. Three additional conditions are constructed based on SQL and the resulting query definition is displayed. Using **ade\_qrygroup()** and the queryIds, (ownerQueryId and tele\_UseQueryId) returned by **ade\_qrydefine()**, conditions can be combined to produce a query which retrieves specific poles within a geographic area as displayed.

```
// Define the location condition.
char* pszJoinOperator    = ""; // none
char* pszBgnCondGrouping = ""; // none
char* pszNotOperator     = ""; // none
char* pszCondType        = "Location";
char* pszEndCondGrouping = ""; // none
```

```

ads_point bufferfencePt1;
bufferfencePt1[X] = 499375;
bufferfencePt1[Y] = 1451421;
ads_point bufferfencePt2;
bufferfencePt2[X] = 499383;
bufferfencePt2[Y] = 1451502;
ads_point bufferfencePt3;
bufferfencePt3[X] = 499422;
bufferfencePt3[Y] = 1451641;
struct resbuf* pLocQueryConditionRb = acutBuildList(
    RTLB,
        RTSTR, "Bufferfence",
        RTSTR, "Inside",
        RTREAL, 100.0,
        RTPOINT, bufferfencePt1,
        RTPOINT, bufferfencePt2,
        RTPOINT, bufferfencePt3,
    RTLE,
    0 );
ade_id locQueryId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pLocQueryConditionRb,
    pszEndCondGrouping);

// Define the first SQL condition.
struct resbuf* pSQL_OwnerRb = acutBuildList(
    RTLB,
        RTSTR, "Pole",
        RTSTR, "OWNER = 'ELECTRIC'",
    RTLE,
    0 );
ade_id OwnerQueryId = ade_qrydefine(
    "AND",
    "",
    "",
    "SQL",
    pSQL_OwnerRb,
    "");

```

```

// Define a second SQL condition.
struct resbuf* pSQLCable_UseRb = acutBuildList(
    RTLB,
    RTSTR, "Pole",
    RTSTR, "CABLE_USE = 'CABLECO'",
    RTLE,
    0 );
ade_id Cable_UseQueryId = ade_qrydefine(
    "OR",
    "",
    "",
    "SQL",
    pSQLCable_UseRb,
    "");

// Define a third SQL condition.
struct resbuf* pSQLTele_UseRb = acutBuildList(
    RTLB,
    RTSTR, "Pole",
    RTSTR, "TELE_USE = 'TELCO'",
    RTLE,
    0 );
ade_id Tele_UseQueryId = ade_qrydefine(
    "OR",
    "",
    "",
    "SQL",
    pSQLTele_UseRb,
    "");

```

The current query definition would appear as the following:

Location; INSIDE BUFFER FENCE

AND SQL:SELECT \* FROM Pole WHERE OWNER = 'ELECTRIC'

OR SQL:SELECT \* FROM Pole WHERE CABLE\_USE = 'CABLECO'

OR SQL:SELECT \* FROM Pole WHERE TELE\_USE = 'TELCO'

```

int resultCode = ade_qrygroup(
    OwnerQueryId,
    Tele_UseQueryId);

```

After the call to [ade\\_qrygroup\(\)](#), the query definition would appear as the following:

```
Location; INSIDE BUFFER FENCE
```

```
AND (SQL:SELECT * FROM Pole WHERE OWNER = 'ELECTRIC'
```

```
OR SQL:SELECT * FROM Pole WHERE CABLE_USE = 'CABLECO'
```

```
OR SQL:SELECT * FROM Pole WHERE TELE_USE = 'TELCO')
```

To ungroup queries, use [ade\\_qryungroup](#).



## ade\_qryhandent

### [Query Functions](#)

---

Gets the entity name for the specified handle.

```
int  
ade_qryhandent(  
    ade_id dwg_id,  
    char* handle,  
    ads_name result);
```

Returns **RTNORM** or an error code.

**dwg\_id**     ID of the drawing in which the object resides.  
**handle**     Original handle of the object in the specified drawing.  
**result**     Output the entity name for the specified drawing ID and handle.

This function provides access to the entity name of an object in a source database.

You must use the retrieved entity name immediately before you call any other function (except **ade\_expreval**) or return control to AutoCAD.

Once you have the entity name of an object, you can use it with other functions. For example, you could use **entget** (or **ads\_entget**) to retrieve the entity and its definition data.

To get the original handle of the object in the source drawing, use the [ade\\_qrygetentlist](#) function.

To obtain a drawing ID, use **ade\_dslist**.

To get the ID of a drawing given a drawing file path, use **ade\_dwggetid**.

The following sample shows how you can combine **ade\_qrygetentlist** and **ade\_qryhandent** to count the number of objects in the source drawing(s) that are of type **lwpolyline**. First the current query definition is cleared and a new location query is defined. The drawing ids for the attached drawings are obtained using **ade\_dslist()**. Entity handles for objects which satisfy the current query are obtained from the active drawing using **ade\_qrygetentlist**. The **resbuf** returned by **ade\_qrygetentlist** is parsed and the entity name for each object is passed to **acdbEntGet()** which returns entity information that is checked for a type of **LWPOLYLINE**. A count of **lwpolyline**s is displayed and all **resbufs** are released as required.

```

int resultCode = ade_qryclear();

char* pszJoinOperator    = ""; // none
char* pszBgnCondGrouping = ""; // none
char* pszNotOperator     = ""; // none
char* pszCondType        = "Location";
char* pszEndCondGrouping = ""; // none

struct resbuf* pLocQueryConditionRb = acutBuildList(
    RTLB,
    RTSTR, "All",
    RTLE,
    0 );
ade_id locQueryId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pLocQueryConditionRb,
    pszEndCondGrouping);

long lwPlineCount = 0;
struct resbuf* pSelectedObjHandlesRb = NULL;
struct resbuf* pEntdata = NULL;

struct resbuf* pAttachedDwgsRb = ade_dslst(ADE_NULLID, 1);
if (NULL != pAttachedDwgsRb) {
    struct resbuf* rb = pAttachedDwgsRb;
    while(NULL != rb) {
        if (ADE_TRUE == ade_dwgisactive(rb->resval.rreal)) {
            pSelectedObjHandlesRb = ade_qrygetentlist(rb->resval.rreal);
            if (NULL != pSelectedObjHandlesRb) {
                struct resbuf* rb1 = pSelectedObjHandlesRb;
                while(NULL != rb1) {
                    ads_name queriedEntity;
                    resultCode = ade_qryhandent(
                        rb->resval.rreal,    // dwg_id
                        rb1->resval.rstring, // handle
                        queriedEntity);
                    pEntdata = acdbEntGet(queriedEntity);
                    while(NULL != pEntdata) {
                        if (0 == pEntdata->restype &&

```

```
        (_tcscmp(pEntdata->resval.rstring, "LWPOLYLINE") == 0)) {

            lwPlineCount++;
        }
        pEntdata = pEntdata->rbnext;
    }
    rb1 = rb1->rbnext;
}
}
acutPrintf(
    "\nThere are %d \"LWPOLYLINES\" in the active drawing set."
    , lwPlineCount);
}
else{
    acutPrintf(
        "\nThe drawing was not active.");
    }
    rb = rb->rbnext;
}
}
else {
    acutPrintf(
        "\nThere are no attached drawings in this project.");
    }
acutRelRb(pAttachedDwgsRb);
acutRelRb(pSelectedObjHandlesRb);
acutRelRb(pEntdata);
```



## ade\_qrylist

### [Query Functions](#)

---

Lists the IDs of the current query conditions.

```
struct resbuf*  
ade_qrylist();
```

Returns a list of the IDs of the current query conditions, or, if there is no current query, **NULL**.

You must release the **resbuf**.

The following sample populates a **resbuf** with query condition id's using **ade\_qrylist()**. Those id's are displayed, and the **resbuf**, is released as required.

```
struct resbuf* pQueryCondIdsRb = ade_qrylist();  
if (NULL != pQueryCondIdsRb){  
    struct resbuf* rb = pQueryCondIdsRb;  
    while(NULL != rb) {  
        acutPrintf(  
            "\nThe following query condition id's have been detected: %.0lf"  
            , rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nNo query conditions were returned for this project.");  
}  
acutRelRb(pQueryCondIdsRb);
```



## ade\_qrysave

### [Query Functions](#)

---

Saves the current query.

```
ade_id  
ade_qrysave(  
    char* catname,  
    struct resbuf* qryparams);
```

Returns the ID of the newly saved query or **ADE\_NULLID**.

**catname**      Category name. The category is created if it does not exist.

**qryparams**    A resbuf list composed of an information type and a value. See the Information Types table below.

### Information Types

<b>name</b>	Query name ( <b>RTSTR</b> )
<b>description</b>	Query description ( <b>RTSTR</b> )
<b>qtype</b>	How the query is saved ( <b>RTSHORT</b> ): 1 = internal (default), 2 = external.
<b>filename</b>	For an external query, full path name ( <b>RTSTR</b> ).
<b>saveoption</b>	Bit code for the save options you are choosing ( <b>RTSHORT</b> ). See the Save Options table below.

The function saves the current query to the current drawing's query library or to a file.

- A query saved to the query library is called an internal query.
- A query saved to a file is called an external query.

You must specify a category name and a query name. In the current drawing and all attached drawings, no two queries can have the same name, even if they are saved in different categories. The default value for a description is the same as the query name. The default value for the storage type is internal. If you want to save the query externally, you must specify a file name for it.

## Save Options

1	Keep reference in query library.
2	Save list of active drawings.
4	Save location coordinates.
8	Save current property alteration definition.
16	Execute automatically.

A query gets a name and an ID only if it is referenced the query library. A new query that you have not yet saved does not have a name or an ID, and neither does an external query unless you keep a reference to it in the query library.

The following sample gets the location of the "QueryFileDirectory" using `ade_prefgetval()`. This path is combined with a query file name to create a parameter representing an external query. A `resbuf` is built which contains the name and description of an internal query plus the information required for the external query. `Ade_qrysave()` is called with with all required parameters, and the returned `queryId` is checked for value with an appropriate message displayed. All `resbufs` are then released, as required.

```
char* pszOptionVar = "QueryFileDirectory";
struct resbuf* pGetPrefValRb = NULL;
pGetPrefValRb = ade_prefgetval(pszOptionVar);
CString sQueryFileName = "Hydro-210.qry";
CString sQueryFilePath = pGetPrefValRb->resval.rstring;
CString sQueryFileLoc = sQueryFilePath + sQueryFileName;
struct resbuf* pQuerySaveParamsRb = acutBuildList(
    RTLB,
        RTSTR, "name", RTSTR, "Hydro-210",
    RTDOTE,
    RTLB,
        RTSTR, "description", RTSTR, "Hydro-210 description",
    RTDOTE,
    RTLB,
        RTSTR, "qtype", RTSHORT, 2,
    RTDOTE,
    RTLB,
        RTSTR, "filename", RTSTR, sQueryFileLoc,
    RTDOTE,
    RTLB,
        RTSTR, "saveoption", RTSHORT, 3,
    RTDOTE,
    RTLE, 0 );
```

```
char* pszQueryCatgryName = "Hydrology";
ade_id queryId = ade_qrysave(pszQueryCatgryName, pQuerySaveParamsRb);
if (queryId != ADE_NULLID) {
    acutPrintf(
        "\nA query has been saved to the internal library: \"%s\"\"",
        , pszQueryCatgryName);
    acutPrintf(
        "\nAn external query has been saved as: %s"
        , sQueryFileLoc);
}
else {
    acutPrintf(
        "\nNo queries have been saved.");
}
acutRelRb(pGetPrefValRb);
acutRelRb(pQuerySaveParamsRb);
```



## ade\_qrysetaltprop

[Query Functions](#)

---

Turns property alteration on or off.

int

```
ade_qrysetaltprop(  
    ade_boolean flag);
```

Returns **RTNORM** or an error code.

**flag** Specifies whether property alteration is on or off: **ADE\_TRUE** = on, **ADE\_FALSE** = off.

This function affects the current query.

If there is no current property alteration definition, this function has no effect. To create a property alteration definition, use [ade\\_altpdefine](#).



## ade\_qrysetcond

### [Query Functions](#)

---

Replaces a query condition.

```
int
ade_qrysetcond(
    ade_id condition_id,
    struct resbuf* condition);
```

Returns **RTNORM** or an error code.

**condition\_id**      Query condition ID to replace.  
**condition**          New query condition (a list). See [ade\\_qrydefine](#).

This function affects the current query.

You cannot alter grouping with this function. Any grouping you specify is ignored. To group or ungroup, use [ade\\_qrygroup](#) or [ade\\_qryungroup](#).

The following sample defines a query using [ade\\_qrydefine\(\)](#). A **resbuf** is created which contains the replacement query condition. This **resbuf** and the query condition id returned by [ade\\_qrydefine\(\)](#) are used as parameters to [ade\\_qrysetcond\(\)](#). The value returned by [ade\\_qrysetcond\(\)](#) is checked against **RTNORM** and an appropriate message is displayed. The **resbuf** is then released as required.

```
char* pszJoinOperator = ""; // none
char* pszBgnCondGrouping = ""; // none
char* pszNotOperator = ""; // none
char* pszCondType = "Property";
char* pszEndCondGrouping = ""; // none
struct resbuf* pQueryConditionRb = acutBuildList(
    RTLB,
    RTSTR, "layer",
    RTSTR, "=",
    RTSTR, "Pond",
    RTLE,
    0 );
```

```

ade_id queryCondId = ade_qrydefine(
    pszJoinOperator,
    pszBgnCondGrouping,
    pszNotOperator,
    pszCondType,
    pQueryConditionRb,
    pszEndCondGrouping);
if (queryCondId != ADE_NULLID) {
    struct resbuf* pSetQueryConditionRb = acutBuildList(
        RTLB,
        RTSTR, "", //joinop
        RTSTR, "", //bgroups
        RTSTR, "", //not_op
        RTSTR, "property",
        RTLB,
        RTSTR, "layer",
        RTSTR, "=",
        RTSTR, "Water",
        RTLE,
        RTLE,
        0 );

    int returnCode = ade_qrysetcond(queryCondId, pSetQueryConditionRb);
    if (RTNORM == returnCode) {
        acutPrintf(
            "\nThe query condition has been modified.");
    }
    else {
        acutPrintf(
            "\nThe query condition has not been modified.");
    }
    acutRelRb(pQueryConditionRb);
    acutRelRb(pSetQueryConditionRb);
}

```



## ade\_qrysetretransform

### [Query Functions](#)

---

Enables or disables transformation for the current report query.

```
int  
ade_qrysetretransform(  
    ade_boolean flag);
```

Returns **RTNORM** or an error code.

**flag**     **TRUE** or **FALSE**, where **TRUE** = transformation enabled, and **FALSE** = transformation disabled.

The following sample enables coordinate transformation for a report query using `ade_qrysetretransform()`.

```
ade_boolean bTxfrmForReportQry = ADE_TRUE;  
int resultCode = ade_qrysetretransform(bTxfrmForReportQry);  
if (RTNORM == resultCode){  
    acutPrintf(  
        "\nReport query transformation has been set.");  
  
}  
else {  
    acutPrintf(  
        "\nNo transformation flag has been set.");  
}
```



## ade\_qrysettype

### [Query Functions](#)

---

Sets the query mode: Preview, Draw, or Report.

```
int
ade_qrysettype(
    char* qrytype,
    ade_boolean multiline,
    char* templ,
    char* filename);
```

Returns **RTNORM** or an error code.

<b>qrytype</b>	Query mode: " <b>preview</b> ", " <b>draw</b> ", or " <b>report</b> ", where " <b>preview</b> " = Display queried objects without retrieving them, similar to Quick View, " <b>draw</b> " = Get queried objects from source drawings or external databases and copy to the current drawing, and " <b>report</b> " = Direct queried information to an output file.
<b>multiline</b>	Whether to write report rows for sub-objects: <b>TRUE</b> or <b>FALSE</b> , where <b>TRUE</b> = Write report rows for sub-objects (objects such as vertices of polylines and attributes of blocks), and <b>FALSE</b> = Write lines for top-level objects only. Relevant only if <b>qrytype</b> is " <b>report</b> ".
<b>templ</b>	Ordered list of object properties to report. Each list element defines a report column. For example, ". <b>type</b> , <b>layer</b> ". Relevant only if <b>qrytype</b> is " <b>report</b> ".
<b>filename</b>	Path and file name of the output file. Relevant only if <b>qrytype</b> is " <b>report</b> ".

This function affects the current query.

The three optional parameters, **multiline**, **templ**, and **filename**, are relevant only if the query mode is "**report**". If the query mode is "**preview**" or "**draw**", omit them.

The following example sets the query mode to **Report** using **ade\_qrysettype()**. The **templ**, (**pszReportTemplate**) parameter represents output of a layer name, and two object data fields.

```
char* pszQueryType = "Report";
char* pszReportTemplate = ".Layer,:AGE_30_49@riblgrp,:AMERI_ES@riblgrp";
char* pszReportFileName = "C:\\ADSRX\\QueryOutput\\Report_010404.txt";
ade_boolean bIsMultiline = ADE_FALSE;
```

```
int resultCode = ade_qrysettype(
    pszQueryType,
    bIsMultiline,
    pszReportTemplate,
    pszReportFileName);
if (RTNORM == resultCode){
    acutPrintf(
        "\nThe report query type has been set.");
}
else {
    acutPrintf(
        "\nUnable to set query type.");
}
```



## ade\_qryungroup

### [Query Functions](#)

---

Ungroups a sequence of two or more query conditions.

```
int  
ade_qryungroup(  
    ade_id condition_id1,  
    ade_id condition_id2);
```

Returns **RTNORM** or an error code.

**condition\_id1**      Condition ID of the first grouped condition.

**condition\_id2**      Condition ID of the last grouped condition.

This function affects the current query.

A query definition consists of a sequence of query conditions. Within such a sequence, there can be subsequences that have been grouped by enclosing them in parentheses. Such groups may have been established when the query was first defined. See the **bggroups** and **endgroups** parameters of [ade\\_qrydefine](#). Or they may have been established afterward by [ade\\_qrygroup](#). However established, you can use **ade\_qryungroup** to undo a group (remove its enclosing parentheses).

When you call **ade\_qryungroup**, the condition you specify as the first of the group (**condition\_id1**) must be a predecessor to the one you specify as the last (**condition\_id2**). The function ungroups the first and the last and any conditions in between.

See the example used in [ade\\_qrygroup](#) for the before and after query definitions resulting from the use of **ade\_qrygroup** and **ade\_qryungroup**.



## ade\_rtdefrange

### [Range Table Functions](#)

---

Defines a range table.

```
ade_id  
ade_rtdefrange(  
    char* tablename,  
    char* description,  
    struct resbuf* range_defn);
```

Returns a range table ID or **ADE\_NULLID**.

<b>tablename</b>	Range table name; can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.
<b>description</b>	Range table description.
<b>range_defn</b>	Range table definition.

A range table allows you to alter properties of queried entities conditionally. It contains a set of property alteration values from which a single value is selected depending on conditions obtaining in the queried entity to be altered.

The **range\_defn** argument is a range table definition, a list of range expressions. Each range expression includes (1) a condition and (2) a property alteration value to return if the condition is true. This information is expressed as a list of three elements: a [range table operator](#) and a comparison value (which together make up the condition), and the return value. You must state each value explicitly. You cannot substitute an expression.

See [Using a Range Table](#) for more information.

The following sample creates a **resbuf** containing the range expression values representing the table definition. **ade\_rtdefrange()** is called with all required parameters, the returned **rangeTableId** is checked for **ADE\_NULLID** and an appropriate message is displayed. Then it releases the **resbuf**, as required.

```
char* pszRangeTblName = "Range_ChgColor";  
char* pszRangeTblDesc = "Change all except red to yellow";  
struct resbuf* pRangeTblDef = acutBuildList(  
    RTLB,
```

```
    RTLB,  
    RTSTR, "=",  
    RTSTR, "1",  
    RTSTR, "1",  
    RTLE,  
    RTLB,  
    RTSTR, "OTHERWISE",  
    RTSTR, "",  
    RTSTR, "2",  
    RTLE,  
    RTLE, 0);
```

```
ade_id rangeTableId = ade_rtdefrange(pszrRangeTblName, pszrRangeTblDesc, pRangeTblDef);  
if (rangeTableId != ADE_NULLID) {  
    acutPrintf(  
        "\nThe range table %s has been created."  
        ,pszrRangeTblName);  
}  
else {  
    acutPrintf(  
        "\nNo range table has been created.");  
}  
acutRelRb(pRangeTblDef);
```



## ade\_rtdeltable

[Range Table Functions](#)

---

Deletes a range table.

int

```
ade_rtdeltable(  
    char* tablename);
```

Returns **RTNORM** or an error code.

**tablename** Range table name; can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.



## ade\_rtgetid

[Range Table Functions](#)

---

Gets a range table ID.

```
ade_id  
ade_rtgetid  
    char* tablename);
```

Returns a range table ID or **ADE\_NULLID**.

**tablename**      Range table name; can be up to 31 characters long. Must be unique, contain no spaces, and start with an alphanumeric character.



## ade\_rtgetprop

### [Range Table Functions](#)

---

Gets the value of a range table property.

```
struct resbuf*
ade_rtgetprop(
    ade_id rt_id,
    char* property);
```

Returns a property value, or list, or **NULL**.

**rt\_id**            Range table ID.

**property**        Property to get the value of. See the Range Table Properties table below.

You must release the **resbuf**.

### Range Table Properties

<b>name</b>	Range table name.
<b>description</b>	Range table description.
<b>expr</b>	Range table definition (list of range expressions).

See [ade\\_rtdefrange](#) for information about setting range table properties.

The following sample obtains a range table id using the sample code associated with [ade\\_rtdefrange\(\)](#). [Ade\\_rtgetprop\(\)](#) uses this range table id along with the specified property parameter and fills the **resbuf** with the corresponding value. If the operation is successful, the **resbuf** is parsed and the requested information is displayed. Then it releases the **resbufs**, as required.

```
char* pszRangeTblName = "Range_ChgColor";
char* pszRangeTblDesc = "Change all except red to yellow";
struct resbuf* pRangeTblDef = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "=",
```

```
        RTSTR, "1",
        RTSTR, "1",
    RTLE,
    RTLB,
        RTSTR, "OTHERWISE",
        RTSTR, "",
        RTSTR, "2",
    RTLE,
    RTLE, 0);
```

```
ade_id rangeTableId = ade_rtdefrange(pszrRangeTblName, pszrRangeTblDesc, pRangeTblDef);
// Range Table has been defined.
```

```
struct resbuf* pRangeTblPropRb = NULL;
char* pszRangeTblPropVal = "description";
if (rangeTableId != ADE_NULLID) {
    pRangeTblPropRb = ade_rtgetprop(rangeTableId, pszRangeTblPropVal);
    if (pRangeTblPropRb != NULL) {
        acutPrintf(
            "\nThe range table \"%s\" property you requested is: \"%s\"."
            , pszRangeTblPropVal, pRangeTblPropRb->resval.rstring);
    }
    else {
        acutPrintf(
            "\nNo range table has been created.");
    }
}
else {
    acutPrintf(
        "\nNo range table has been created.");
}
acutRelRb(pRangeTblDef);
acutRelRb(pRangeTblPropRb);
```



## ade\_rtolist

### [Range Table Functions](#)

---

Lists the IDs of all range tables defined in the project.

```
struct resbuf*  
ade_rtolist();
```

Returns a list of range table IDs or **NULL**.

You must release the **resbuf**.

The following sample parses the **resbuf** returned by **ade\_rtolist()** and prints a list of range tables in the current project. Then it releases the **resbuf**, as required.

```
struct resbuf* pRtListRb = NULL;  
pRtListRb = ade_rtolist();  
if (NULL != pRtListRb) {  
    int nRangeTables = 0;  
    struct resbuf* rb = pRtListRb;  
    while(rb != NULL) {  
        ++nRangeTables;  
        acutPrintf(  
            "\nRange table %d has the Id: %.0lf", nRangeTables, rb->resval.rreal);  
        rb = rb->rbnext;  
    }  
}  
else {  
    acutPrintf(  
        "\nThere are no range tables in this project.");  
}  
acutRelRb(pRtListRb);
```



## ade\_saveobjs

### [Object Saving Functions](#)

---

Saves objects queued for saving back to the source drawings.

```
int  
ade_saveobjs(  
    struct resbuf* priorities);
```

Returns **RTNORM** or an error code.

**priorities** List containing one to four save back operation codes, depending on the number of save back operations you are specifying. List the codes in order of their relative priority. See the Save-Back Operation Codes table below.

### Save-Back Operation Codes

Code	Operation	Description
1	Came From	Saves objects to their source drawings.
2	Selective	Saves objects you select to the drawings you specify.
3	Layer	Saves objects to layers in the source drawings that use the same names as those in which the objects lie.
4	Area	Saves objects to the source drawing within whose extents they lie, even if only partially within.

During the save back operation, the options are executed in the order specified in the **priorities** list. The list must contain at least one option. For example:

The following example creates a **resbuf** containing save-back operation code(s). **Ade\_saveobjs()** uses this **resbuf** and returns **RTNORM** if successful. Status messages are displayed based on the returned code. Then it releases the **resbuf**, as required.

```
struct resbuf* pSaveBackPriorityRb = acutBuildList(RTSHORT, 1, 0);  
int returnCode = ade_saveobjs(pSaveBackPriorityRb);  
if (RTNORM == returnCode) {
```

```
    acutPrintf(
        "\nObjects were successfully saved.");
}
else {
    acutPrintf(
        "\nObjects were not saved.");
}
acutRelRb(pSaveBackPriorityRb);
```

To save a selection set to a specific drawing, use [ade\\_savetodwg](#).



## ade\_savetodwg

### [Object Saving Functions](#)

---

Saves a selection set to a specific drawing.

```
int  
ade_savetodwg(  
    ads_name sel_set,  
    ade_id dwg_id);
```

Returns **RTNORM** or an error code.

**sel\_set**      Selection set name.  
**dwg\_id**      Drawing ID of the destination drawing.

To save objects queued for saving back to the source drawings, use [ade\\_saveobjs](#).



## ade\_sqlgetenvstring

[SQL Environment Functions](#)

---

Gets a string describing the SQL environment.

```
char*
ade_sqlgetenvstring(
    char* linktemplate);
```

Returns a string describing the SQL environment or **NULL**.

**linktemplate**      Link template.

For more information about link templates using SQL, see the AutoCAD online documentation.

The following sample gets the SQL environment string using **ade\_sqlgetenvstring()** and displays that information based on the returned value.

```
char* pszLinkTemplateName = "BlockGroup";
char* pszSQLEnvironmentString = ade_sqlgetenvstring(pszLinkTemplateName);
if (NULL == pszSQLEnvironmentString ||
    _tcscmp(pszSQLEnvironmentString, "") == 0) {

    acutPrintf(
        "\nNo environment string was obtained.");
}
else {
    acutPrintf(
        "\nThe current environment string is: \"%s\"."
        , pszSQLEnvironmentString);
}
```



## ade\_userget

[User Security Functions](#)

---

Gets the login name or entity lock name of the local user.

```
int  
ade_userget(  
    ade_boolean for_entity_locks,  
    char* username);
```

Returns **RTNORM** or an error code.

**for\_entity\_locks**      What to do if the local user is not logged into the application (optional): **TRUE** or **FALSE**, where **TRUE** = Get the user name used to identify the owner of object locks set locally, and **FALSE** = Return **FALSE**. Omitting this argument is the same as supplying **FALSE**.

**username**              The user name or the empty string.

A user name can have as many as 32 characters.

If the local user is logged into the application, the user's application login name is used to identify the owner of object locks set locally. If the local user is not logged into the application, the user's operating system login name is used.

This function helps you determine if the owner of a particular object lock is the local user.



## ade\_usergetrights

[User Security Functions](#)

---

Gets the access rights of a user.

```
int  
ade_usergetrights(  
    char* userName,  
    int* pUserRights);
```

Returns **RTNORM** or an error code.

**username** Login name, at most 32 characters.

**pUserRights** An integer equal to the sum of the rights allowed.

If the **userName** argument is **NULL**, the function returns the rights of the current user.

The return value is a bit code for the rights allowed, as shown in the following table.

### User Rights Codes

Code	User Rights
1	Superuser (in which case the other bits don't matter).
2	Permission to alter the drawing set.
4	Permission to edit objects.
8	Permission to execute a draw query.
16	Permission to edit Feature Class definition.

If the **username** argument is omitted or **NULL**, and there is no current user, the function returns a bit code with all bits set, because the no-current-user condition is possible only if the system option "**ForceUserLogin**" is set to **NULL**, in which case all users have all rights except those reserved for a superuser.

Only a superuser can specify a login name other than their own. If the **username** argument is not the login

name of the current user, and the current user does not have superuser rights, the function returns **NULL**, and the message "Access is denied" is added to the error stack.



## ade\_userlist

### [User Security Functions](#)

---

Lists the current users.

```
struct resbuf*  
ade_userlist();
```

Returns a list of user login names or **NULL**.

The following sample parses the **resbuf** returned by **ade\_userlist()** and prints a list of user names. Then it releases the **resbuf**, as required.

```
struct resbuf* pUserListRb = NULL;  
pUserListRb = ade_userlist();  
if (NULL != pUserListRb) {  
    while (pUserListRb != NULL) {  
        pUserListRb = pUserListRb->rbnext;  
        acutPrintf(  
            "\nUser - %s",pUserListRb->resval);  
    }  
} else {  
    acutPrintf(  
        "\nUser list could not be created");  
}  
acutRelRb(pUserListRb);
```



## ade\_userset

[User Security Functions](#)

---

Logs in a user.

```
int  
ade_userset(  
    char* username,  
    char* password);
```

Returns **RTNORM** or an error code

**username**      Login name or **NULL**.  
**password**      Password or **NULL**

If either argument is omitted, the User Login dialog box displays. If a login name was specified, it appears in the dialog's Login Name field.

If both arguments are specified, but the user cannot be logged in, one of the following messages is added to the error stack:

- Invalid user name.
- Invalid password.

If the drawing set includes active drawings containing locked entities, the current user cannot be changed. If you attempt to log in a different user under those conditions, the following error message is added to the error message stack:

- Cannot login again when drawings are locked/active.



## ade\_usersetrights

### [User Security Functions](#)

---

Sets the access rights for a user.

```
int  
ade_usersetrights(  
    char* userName,  
    int userRights);
```

Returns **RTNORM** or an error code.

**userName**      Login name.

**userRights**    An integer equal to the sum of the rights to allow. See the User Rights Codes table below.

#### User Rights Codes

Code	User Rights
1	Superuser (in which case the other bits don't matter).
2	Permission to alter the drawing set.
4	Permission to edit objects.
8	Permission to execute a draw query.
16	Permission to edit Feature Class definition.

This function cannot execute unless the current user has superuser rights, and it cannot change the rights of the current user in any case. If an ordinary user is logged in when this function is called, or a superuser is logged in and the function call would change the rights of the current user, the function returns an error code of **-1002**, and the following message is added to the error message stack:

Access denied



## ade\_version

[Other Functions](#)

---

Gets the version number of the Data Extension programming interface.

```
struct resbuf*  
ade_version();
```

Returns a version number (**string**) or **NULL**.

You must release the **resbuf**.

The following sample parses the **resbuf** returned by **ade\_version()** and prints the version number. Then it releases the **resbuf**, as required.

```
struct resbuf* pVersionRb = NULL;  
pVersionRb = ade_version();  
if (NULL != pVersionRb) {  
    acutPrintf(  
        "\nThe current version is: %s", pVersionRb->resval);  
}  
else {  
    acutPrintf(  
        "\nThe version could not be determined");  
}  
acutRelRb(pVersionRb);
```



## Drawing Functions

### [Data Extension Function Synopsis](#)

---

The functions for drawing management begin with `ade_dwg`.

<a href="#">ade_dwgactivate</a>	Activates a drawing.
<a href="#">ade_dwgactualpath</a>	Returns the full path of a drawing.
<a href="#">ade_dwgaliaspath</a>	Returns the alias path of a drawing.
<a href="#">ade_dwgattriblist</a>	Returns a list of attribute tags for the specified block name.
<a href="#">ade_dwgdeactivate</a>	Deactivates a drawing.
<a href="#">ade_dwggetid</a>	Gets the drawing ID of a drawing.
<a href="#">ade_dwggetsetting</a>	Gets a drawing setting value.
<a href="#">ade_dwghaslocks</a>	Checks if a drawing has locked objects.
<a href="#">ade_dwgindex</a>	Applies specified index operations to a drawing.
<a href="#">ade_dwgindexdef</a>	Specifies which indexes are to be created or removed.
<a href="#">ade_dwgisactive</a>	Checks if a drawing is active.
<a href="#">ade_dwgistoplevel</a>	Checks if a drawing is directly attached to the project drawing.
<a href="#">ade_dwgproplist</a>	Lists all values found in a drawing for a given drawing property.
<a href="#">ade_dwgquickview</a>	Displays a quick view of a drawing.
<a href="#">ade_dwgselectdlg</a>	Displays the Select Drawings dialog box.
<a href="#">ade_dwgsetof</a>	Identifies the drawings to which a given drawing is attached.
<a href="#">ade_dwgsetsetting</a>	Sets a drawing setting value.
<a href="#">ade_dwgunlock</a>	Removes all object locks from a drawing.
<a href="#">ade_dwgzoomextents</a>	Zooms to the extents of the active drawings.



## Drawing Set Functions

### [Data Extension Function Synopsis](#)

---

The functions for drawing set management begin with `ade_ds`.

<a href="#">ade_dsattach</a>	Attaches a drawing to the project.
<a href="#">ade_dsdetach</a>	Detaches a drawing from the project.
<a href="#">ade_dsisnested</a>	Checks if a drawing has nested drawings.
<a href="#">ade_dslist</a>	Lists the drawings attached to a given drawing.
<a href="#">ade_dsproplist</a>	Lists all values found in the drawing set for a given drawing property.



## Drive Alias Functions

[Data Extension Function Synopsis](#)

---

The drive alias functions begin with `ade_alias`.

<a href="#">ade_aliasadd</a>	Creates a drive alias.
<a href="#">ade_aliasdelete</a>	Deletes a drive alias.
<a href="#">ade_aliasgetlist</a>	Lists all drive aliases in the project.
<a href="#">ade_aliasupdate</a>	Assigns a new drive and path to a drive alias.



## Error Message Functions

### [Data Extension Function Synopsis](#)

---

The functions for handling error messages begin with `ade_err`.

<a href="#">ade_errclear</a>	Clears the error stack.
<a href="#">ade_errcode</a>	Gets the error code for a specific error on the error stack.
<a href="#">ade_errgetlevel</a>	Gets the system error level.
<a href="#">ade_errmsg</a>	Gets the error message for a specific error on the error stack.
<a href="#">ade_errpush</a>	Pushes an error message to the stack.
<a href="#">ade_errpushstatement</a>	Pushes a statement to the stack.
<a href="#">ade_errqty</a>	Returns the number of error messages on the stack.
<a href="#">ade_errsetlevel</a>	Sets the system error level.
<a href="#">ade_errshowdlg</a>	Displays the Map Messages dialog box, which shows a list of error messages on the stack.
<a href="#">ade_errstatement</a>	Gets the erroneous statement for a specific error on the stack.
<a href="#">ade_errtype</a>	Gets the type of a specific error in the stack.



## Expression Evaluation Function

[Data Extension Function Synopsis](#)

---

The expression evaluation function begins with `ade_exp`.

---

[ade\\_expreval](#)

Evaluates an AutoCAD Map expression.

---



## Object Data Functions

### [Data Extension Function Synopsis](#)

---

The functions for object data management begin with `ade_od`.

<a href="#">ade_odaddfield</a>	Adds fields to a table.
<a href="#">ade_odaddrecord</a>	Attaches data to an object.
<a href="#">ade_odattachrecord</a>	Attaches a new record to an object.
<a href="#">ade_oddefinetab</a>	Creates an object data table.
<a href="#">ade_oddeletefield</a>	Deletes fields from a table.
<a href="#">ade_oddeletetab</a>	Deletes a table.
<a href="#">ade_oddelrecord</a>	Deletes a record.
<a href="#">ade_odfreerec</a>	Frees the memory claimed in defining a new record.
<a href="#">ade_odgetfield</a>	Gets a field value.
<a href="#">ade_odgetrecfield</a>	Gets a field value using a record ID.
<a href="#">ade_odgetrecord</a>	Gets a record ID.
<a href="#">ade_odgettables</a>	Lists the tables attached to an object.
<a href="#">ade_odmodifyfield</a>	Modifies field properties in a table.
<a href="#">ade_odmodifytab</a>	Redefines a table.
<a href="#">ade_odnewrecord</a>	Defines a new object data record.
<a href="#">ade_odpresetfield</a>	Assigns a value to a field in a new record.
<a href="#">ade_odrecordqty</a>	Counts the records attached to an object.
<a href="#">ade_odsetfield</a>	Sets a field value.
<a href="#">ade_odtabledefn</a>	Gets a table definition.

[ade\\_odtablelist](#)

Lists the tables in the project.

---



## Object Editing Functions

### [Data Extension Function Synopsis](#)

---

The functions for object editing begin with `ade_edit`.

<a href="#">ade_editdefcen</a>	Defines a new label point for an object.
<a href="#">ade_editlockederased</a>	Gets the objects in the save set that have been erased.
<a href="#">ade_editislocked</a>	Gets lock information about an object if it is locked.
<a href="#">ade_editlocked</a>	Gets the objects in the save set that have been modified.
<a href="#">ade_editlockobjs</a>	Locks a set of objects and adds them to the save set.
<a href="#">ade_editnew</a>	Gets the objects in the save set that are new.
<a href="#">editunlockobjs</a>	Unlocks a set of objects and removes them from the save set.



## Object Saving Functions

[Data Extension Function Synopsis](#)

---

The object saving functions begin with `ade_save`.

<a href="#">ade_saveobjs</a>	Saves objects queued for saving back to the source drawings.
<a href="#">ade_savetodwg</a>	Saves a selection set to a specific drawing.



## Option Functions

### [Data Extension Function Synopsis](#)

---

The option functions begin with `ade_pref`.

<a href="#">ade_prefgetval</a>	Gets an option setting.
<a href="#">ade_prefsetval</a>	Sets an option.



## Property Alteration Functions

### [Data Extension Function Synopsis](#)

---

The functions for specifying how objects retrieved in a query should be altered begin with `ade_`altp.

<a href="#">ade_altpclear</a>	Clears the current property alteration definition.
<a href="#">ade_altpdefine</a>	Creates a property alteration expression.
<a href="#">ade_altpdelprop</a>	Deletes a property alteration expression.
<a href="#">ade_altpgetprop</a>	Gets a property alteration expression.
<a href="#">ade_altplist</a>	Lists the IDs of the current property alteration expressions.
<a href="#">ade_altpsetprop</a>	Modifies a property alteration expression.



## Query Functions

### [Data Extension Function Synopsis](#)

---

The functions for query management begin with `ade_qry`.

<a href="#">ade_qryclear</a>	Clears the current query.
<a href="#">ade_qrydefine</a>	Defines a query.
<a href="#">ade_qryexecute</a>	Executes the current query.
<a href="#">ade_qrygetcond</a>	Gets a query condition.
<a href="#">ade_qrygetdwgandhandle</a>	Gets the source drawing ID and original handle of a queried object.
<a href="#">ade_qrygetentlist</a>	Returns entity handles for objects that satisfy the current query.
<a href="#">ade_qrygetretransform</a>	Checks whether transformation is enabled for the current report query.
<a href="#">ade_qrygroup</a>	Groups query conditions.
<a href="#">ade_qryhandent</a>	Gets the entity name for the specified handle.
<a href="#">ade_qrylist</a>	Lists the IDs of the current query conditions.
<a href="#">ade_qrysave</a>	Saves the current query.
<a href="#">ade_qrysetaltprop</a>	Turns property alteration on or off.
<a href="#">ade_qrysetcond</a>	Modifies a query condition.
<a href="#">ade_qrysetretransform</a>	Enables or disables transformation for the current report query.
<a href="#">ade_qrysettype</a>	Sets the query mode: Preview, Draw, or Report.
<a href="#">ade_qryungroup</a>	Ungroups query conditions.



## Query Library Functions

### [Data Extension Function Synopsis](#)

---

The functions for query library management begin with `ade_ql`.

<a href="#">ade_qldelctgy</a>	Deletes a query library category.
<a href="#">ade_qldelquery</a>	Deletes a query from the query library.
<a href="#">ade_qlgetctgyinfo</a>	Gets information about a query category.
<a href="#">ade_qlgetqryinfo</a>	Gets information about a query.
<a href="#">ade_qllistctgy</a>	Lists the query category IDs.
<a href="#">ade_qlloadqry</a>	Makes a saved query current.
<a href="#">ade_qlqrygetid</a>	Gets a query ID.
<a href="#">ade_qlsetctgyname</a>	Changes a query category name.
<a href="#">ade_qlsetquery</a>	Changes a query name, description, or the category it belongs to.



## Range Table Functions

[Data Extension Function Synopsis](#)

---

The functions for range table management begin with `ade_rt`.

<a href="#">ade_rtdefrange</a>	Defines a range table.
<a href="#">ade_rtdeltable</a>	Deletes a range table.
<a href="#">ade_rtgetid</a>	Gets a range table ID.
<a href="#">ade_rtgetprop</a>	Gets the value of a range table property.
<a href="#">ade_rtolist</a>	Lists the IDs of all range tables defined in the project.



## SQL Environment Functions

### [Data Extension Function Synopsis](#)

---

The SQL environment functions begins with `ade_sql` or `ade_key`.

<a href="#">ade_keycolumnlist</a>	Returns a list of the key column names for the specified link path name.
<a href="#">ade_sqlgetenvstring</a>	Gets a string describing the SQL environment.



## User Security Functions

### [Data Extension Function Synopsis](#)

---

The user security functions begin with `ade_user`.

<a href="#">ade_userget</a>	Gets the login name or entity lock name of the local user.
<a href="#">ade_usergetrights</a>	Gets the access rights of the specified user.
<a href="#">ade_userlist</a>	Lists the current users.
<a href="#">ade_userset</a>	Logs in a user.
<a href="#">ade_usersetrights</a>	Sets the access rights for the specified user.



## Other Functions

### [Data Extension Function Synopsis](#)

---

Miscellaneous data extension functions.

<a href="#">ade_entsetlocation</a>	Sets a new text-label point for a drawing object.
<a href="#">ade_osfexpand</a>	Searches a directory and returns a list of file names.
<a href="#">ade_version</a>	Gets the version number of the Data Extension API.

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: eAnnotationExpressionFields Enumeration

[AcMapAnnotationManager Namespace](#)

Enumerates annotation expression fields.

```
enum eAnnotationExpressionFields {  
    kUndefinedField = 0x00,  
    kAttDefAnnotationString = 0x01,  
    kAttDefTextStyle = 0x02,  
    kAttDefTextHeight = 0x03,  
    kAttDefTextLayer = 0x04,  
    kAttDefTextColor = 0x05,  
    kAttDefTextRotation = 0x06,  
    kAttDefTextJustification = 0x07,  
    kAttDefTextLineWeight = 0x08,  
    kAttDefTextTrueColor = 0x09,  
    kBlockPosition = 0x10,  
    kBlockRotation = 0x20,  
    kBlockScaleFactor = 0x30,  
    kBlockColor = 0x40,  
    kBlockLayer = 0x50,  
    kBlockLinetype = 0x60,  
    kBlockLineWeight = 0x70,  
    kBlockAssocEnt = 0x80,  
    kBlockTrueColor = 0x90  
};
```

};

File

AcMapAnnotationManager.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: eInsertPrecedenceMode Enumeration

[AcMapAnnotationManager Namespace](#)

Enumerates annotation insert precedence modes.

```
enum eInsertPrecedenceMode {  
    kExpressionPriority = 0,  
    kOverridePriority  
};  
File
```

AcMapAnnotationManager.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::

AnnotationBlockReferenceAssociatedObjectId Function

[AcMapAnnotationManager Namespace](#)

Returns the object ID of the entity that an annotation reference references.

```
AcDbObjectId AnnotationBlockReferenceAssociatedObjectId(  
    const AcDbBlockReference* pBlkRef  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pBlkRef	Input annotation reference to examine.

Returns

Returns the object ID of the associated entity, or AcDbObjectId::kNull if an error occurs.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::CreateAnnotationText Function

[AcMapAnnotationManager Namespace](#)

Creates a new annotation text object in a template.

```
Acad::ErrorStatus CreateAnnotationText(  
    AcDbObjectId & newAnnotationTextId,  
    AcDbObjectId templateId  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newAnnotationTextId	Output object ID of the newly created annotation text object.
templateId	Input object ID of the annotation template to add the annotation text object to.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

The new annotation text object will have default property values (that is, no expression overrides). At minimum, you should set the new object's annotation string (kAttDefAnnotationString expression) immediately after creation, typically with SetExpressionString().

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReference Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to an associated entity.

```
Acad::ErrorStatus InsertAnnotationReference(  
    AcDbObjectId& newBlockReferenceId,  
    AcDbObjectId templateOrBlockId,  
    const AcDbObjectId assocEnt,  
    const AcDbBlockReference * pMatchThisBlockReference = NULL  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceId	Input/Output object ID of the newly created annotation reference. If newBlockReferenceId is AcDbObjectId::kNull, then a new annotation reference is created. If not, then the AcDbBlockReference whose Object ID is newBlockReferenceID is reused.
templateOrBlockId	Input object ID of an annotation template's block table record or a regular AutoCAD block table record to attach.
assocEnt	Input object ID of the entity to which the reference will be attached.
pMatchThisBlockReference	Input optional pointer to an existing annotation reference, whose static-property and expression-string overrides are used to create the new annotation reference.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

This function takes a template or block table record ID and an associated entity object ID. The new block reference object ID is an output parameter. If the caller provides an optional pointer to an existing annotation reference, that reference's static-property and expression-string overrides are used to create the new annotation reference.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReference Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to an associated entity.

```
Acad::ErrorStatus InsertAnnotationReference(  
    AcDbObjectId& newBlockReferenceId,  
    AcDbObjectId templateOrBlockId,  
    const AcDbObjectId assocEnt,  
    const AnnotationOverrides & overrides  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceId	Input/Output object ID of the newly created annotation reference. If newBlockReferenceId is AcDbObjectId::kNull, then a new annotation reference is created. If not, then the AcDbBlockReference whose Object ID is newBlockReferenceID is reused.
templateOrBlockId	Input object ID of an annotation template's block table record or a regular AutoCAD block table record to attach.
assocEnt	Input object ID of the entity to which the reference will be attached.
overrides	Input AnnotationOverridesstruct containing the static-property and expression-string overrides to be used when creating the new annotation reference. In the AnnotationOverrides structure, pass a NULL for each individual property that you do not want to override; instead, these property values will be computed from the template's corresponding expression or static property.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

## Remarks

This function takes a template or block table record ID and an associated entity object ID. The new block reference object ID is an output parameter. The static-property and expression-string overrides of the input structure are used to create the new annotation reference (unless a NULL prevents an override).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::InsertAnnotationReferences Function

[AcMapAnnotationManager Namespace](#)

Attaches an annotation reference to one or more associated entities.

```
Acad::ErrorStatus InsertAnnotationReferences(  
    AcDbObjectIdArray& newBlockReferenceIds,  
    AcDbObjectId templateOrBlockId,  
    const AcDbObjectIdArray& assocEntArray,  
    const AcDbBlockReference * pMatchThisBlockReference = NULL  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
newBlockReferenceIds	Output array of object IDs of the newly created annotation references.
templateOrBlockId	Input object ID of an annotation template's block table record or a regular AutoCAD block table record to attach.
assocEntArray	Input array of object IDs of the entities to which the references will be attached.
pMatchThisBlockReference	Input optional pointer to an existing annotation reference, whose static-property and expression-string overrides are used to create the new annotation reference.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Remarks

This function takes a template or block table record ID and an array of associated entity object IDs. An array of new block reference object IDs is an output parameter. If the caller provides an optional pointer to an existing annotation reference, that reference's static-property and expression-string

overrides are used to create the new annotation references.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: IsAnnotationBlockReference Function

[AcMapAnnotationManager Namespace](#)

Determines whether a block reference is an annotation reference.

```
bool IsAnnotationBlockReference(  
    const AcDbBlockReference * pBlkRef  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pBlkRef	Input block reference to examine.
Returns	

Returns true if the block reference is an annotation reference; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: IsAnnotationText Function

[AcMapAnnotationManager Namespace](#)

Determines whether an attribute definition is an annotation text entity.

```
bool IsAnnotationText(  
    const AcDbAttributeDefinition * pAttDef  
);  
File
```

AcMapAnnotationManager.h

Parameters	Description
pAttDef	Input attribute definition to examine.
Returns	

Returns true if the attribute definition is an annotation text entity; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetAnnotationInsertPrecedenceMode Function

[AcMapAnnotationManager Namespace](#)

Sets the annotation insert precedence mode.

```
AcMapAnnotationManager::eInsertPrecedenceMode SetAnnotationInsertPre  
AcMapAnnotationManager::eInsertPrecedenceMode newPrecedence
```

```
);
```

```
File
```

AcMapAnnotationManager.h

Parameters	Description
newPrecedence	New annotation insert precedence mode to set.

Returns

Returns the previous precedence mode.

Remarks

Annotation references may give expressions priority (legacy behavior) or may give overrides priority. The previous insert mode is returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: SetExpressionString Function

[AcMapAnnotationManager Namespace](#)

Sets an expression string to store with an annotation template, annotation reference, or annotation text object.

```
Acad::ErrorStatus SetExpressionString(  
    AcMapExpression & pExpression,  
    AcDbObject* pObj,  
    AcMapAnnotationManager::eAnnotationExpressionFields field  
);
```

File

AcMapAnnotationManager.h

Parameters	Description
pExpression	Input expression to store with the object.
pObj	Input pointer to the AcDbBlockTableRecord (template), AcDbBlockReference (reference), or AcDbAttributeDefinition (text) to store the expression string with.
field	Input <a href="#">eAnnotationExpressionFields</a> value of the expression to store. For annotation templates and annotation references, use only kBlock* codes; for annotation text entities, use only kAttDef* codes.

Returns

Returns Acad::eOk if successful; otherwise, returns an error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager::TemplateNameBTRPrefix Function

[AcMapAnnotationManager Namespace](#)

Returns the prefix that is prepended to all annotation block table records internally.

```
const ACHAR* TemplateNameBTRPrefix();
```

File

AcMapAnnotationManager.h

Returns

Returns the internal prefix prepended to all annotation block table records.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapAnnotationManager Namespace](#)

AcMapAnnotationManager:: AnnotationOverrides Structure

[AcMapAnnotationManager Namespace](#)

Structure that holds all static-property and expression-string overrides used for creating or updating annotations.

```
struct AnnotationOverrides {  
    const double * pdRotationOverride;  
    const double * pdScaleFactorOverride;  
    const AcCmColor * pColorOverride;  
    const ACHAR * pszLayerOverride;  
    const ACHAR * pszLinetypeOverride;  
    const AcDb::Lineweight * pLineweightOverride;  
    const ACHAR * pszRotationExpressionOverride;  
    const ACHAR * pszScaleFactorExpressionOverride;  
    const ACHAR * pszColorExpressionOverride;  
    const ACHAR * pszLayerExpressionOverride;  
    const ACHAR * pszLinetypeExpressionOverride;  
    const ACHAR * pszLineweightExpressionOverride;  
    const ACHAR * pszPositionExpressionOverride;  
};
```

File

AcMapAnnotationManager.h

Parameters	Description
pdRotationOverride	An override block rotation value, or NULL to override.
pdScaleFactorOverride	An override block scale factor value, or NULL not to override.
pColorOverride	An override block color value, or NULL not to override.
pszLayerOverride	An override block layer value, or NULL not to override.
pszLinetypeOverride	An override block linetype value, or NULL not to override.
pLineweightOverride	An override block lineweight value, or NULL no not to override.

pszRotationExpressionOverride	An override block rotation expression, or NULL not to override.
pszScaleFactorExpressionOverride	An override block scale factor expression, or NULL not to override.
pszColorExpressionOverride	An override block color expression, or NULL not to override.
pszLayerExpressionOverride	An override block layer expression, or NULL not to override.
pszLinetypeExpressionOverride	An override block linetype expression, or NULL not to override.
pszLineweightExpressionOverride	An override block lineweight expression, or NULL not to override.
pszPositionExpressionOverride	An override block position expression, or NULL not to override.

#### Remarks

Set a parameter to NULL if you do not want to override the default or existing value of that property.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: ~AcMapClassificationManager Destructor  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Destroys an instance of this class.

```
virtual ~AcMapClassificationManager();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: AcMapClassificationManager Constructor  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Constructs an instance of this class by using the specified AutoCAD database.

```
AcMapClassificationManager(  
    AcDbDatabase * pDb  
);
```

Parameters	Description
pDb	Input AutoCAD database.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)

AcMapClassificationManager:: Audit Method

[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Examines an entity for out-of-range or missing classified-property values, fixing these values if desired.

```
AcMapObjClass::EErrCode Audit(  
    AcDbObjectId& entId,  
    bool bFixOutOfRange,  
    bool bFixMissingProperties  
);
```

Parameters	Description
entId	Input ID of the entity to examine.
bFixOutOfRange	Input true to reset each out-of-range property to its default value, or false to not reset these properties.
bFixMissingProperties	Input true to add each missing property to the entity, or false to not add these properties.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eEntityNotClassified if the entity is unclassified. Returns [AcMapObjClass::EErrCode](#) eOutOfRange if at least one property value is out of range. This error code is returned only if bFixOutOfRange is true and an out-of-range property could not be fixed. If the range of the Layer property specifies a layer that does not exist in the current drawing, for example, attempting to fix this nonexistent layer will fail. Returns [AcMapObjClass::EErrCode](#) eMissingProperty if the entity is missing at least one property. This error code is returned only if bFixMissingProperties is true and a missing property could not be added to the entity. If a missing object-data property refers to a table and column with the same name but different type as the one in the current drawing, for example, the type mismatch prevents adding the property and setting its value. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this

message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: Audit Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Examines multiple entities for out-of-range or missing classified-property values, listing and fixing these values if desired.

```
AcMapObjClass::EErrCode Audit(  
    AcDbObjectIdArray* paEntIdsOutOfRange,  
    AcDbObjectIdArray* paEntIdsMissingProperties,  
    AcDbObjectIdArray* paEntIdsNotClassified,  
    AcDbObjectIdArray& aEntIds,  
    bool bFixOutOfRange,  
    bool bFixMissingProperties  
);
```

Parameters	Description
paEntIdsOutOfRange	Output entity IDs of out-of-range properties, or NULL if not needed by the caller.
paEntIdsMissingProperties	Output entity IDs of missing properties, or NULL if not needed by the caller.
paEntIdsNotClassified	Output IDs of unclassified entities. or NULL if not needed by the caller.
aEntIds	Input IDs of the entities to examine.
bFixOutOfRange	Input true to reset each out-of-range property to its default value, or false to not reset these properties.
bFixMissingProperties	Input true to add each missing property to the entity, or false to not add these properties.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for at least one entity.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: CanCurrentUserAlterSchema Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Determines whether the current user has sufficient privileges to change the classification information in the feature-definition file.

```
bool CanCurrentUserAlterSchema() const;
```

Returns

Returns true if the user has adequate privileges; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)

[AcMapClassificationManager:: ClearAllTags Method](#)

[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Unclassifies an entity entirely. Unclassifying entities does not trigger internal transactions. This function differs from `Unclassify()` because it ignores the feature-definition file that was used to classify the entity.

```
AcMapObjClass::EErrCode ClearAllTags(  
    AcDbObjectId& entId  
);
```

Parameters

Description

entId

Input ID of the entity to unclassify entirely.

Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if successful. Returns [AcMapObjClass::EErrCode](#) `eAlreadyUnclassified` if the entity is already unclassified. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)

[AcMapClassificationManager:: ClearAllTags Method](#)

[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Unclassifies multiple entities entirely. Unclassifying entities does not trigger internal transactions. This function differs from `Unclassify()` because it ignores the feature-definition file that was used to classify the entities.

```
AcMapObjClass::EErrCode ClearAllTags(  
    AcDbObjectIdArray* paAlreadyClearedIds,  
    AcDbObjectIdArray& aEntIds  
);
```

Parameters	Description
<code>paAlreadyClearedIds</code>	Output IDs of the entities that had no classification information at all, or NULL if not needed by the caller.

<code>aEntIds</code>	Input IDs of the entities to unclassify entirely.
----------------------	---

## Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if all entities are cleared successfully. Returns [AcMapObjClass::EErrCode](#) `eAlreadyUnclassified` if at least one entity is already unclassified. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for at least one entity.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::DetachCurrentFeatureDefinitionFile Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Detaches the current feature-definition file from the current drawing.

[AcMapObjClass::EErrCode](#) DetachCurrentFeatureDefinitionFile();

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns  
[AcMapObjClass::EErrCode](#) eFailed if the process failed.

Remarks

This function succeeds even if no feature-definition file is attached to the current drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetAllTags Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the classification tags of an entity.

```
AcMapObjClass::EErrCode GetAllTags(  
    AcMapStringArray& aTagNames,  
    AcMapStringArray& aSchemaNames,  
    const AcDbObjectId& entId  
) const;
```

Parameters	Description
aTagNames	Output array of feature class names.
aSchemaNames	Output array of feature-definition file names. The array index of each feature-definition file name matches the index of each corresponding feature class name in aTagNames.
entId	Input ID of the entity to examine.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eEntityNotClassified if the entity is unclassified. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

If an entity was classified multiple times by using different feature-definition files, this function retrieves all the entity's feature class names.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetClassifiedEntities Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the classified entities in the current drawing.

```
bool GetClassifiedEntities(  
    AcDbObjectIdArray& aEntIds  
) const;
```

Parameters	Description
aEntIds	Output IDs of classified entities.

## Returns

Returns true if at least one entity ID is returned; otherwise, returns false if no entity ID is returned or the process failed.

## Remarks

An entity is considered to be classified even if the corresponding feature class definition is not in the attached feature-definition file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetFeatureClassDefinitionCount Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Counts the number of feature class definitions in the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrorCode GetFeatureClassDefinitionCount(  
    long* pICount  
) const;
```

Parameters	Description
pICount	Output number of class definitions.
Returns	

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetFeatureClassNames Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the feature class definition names in the feature-definition file attached to the current drawing.

```
AcMapObjClass::EErrorCode GetFeatureClassNames(  
    AcMapStringArray& aClassNames  
) const;
```

Parameters	Description
aClassNames	Output array of class definition names.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetProperties Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the properties and their values of an entity.

```
AcMapObjClass::EErrCode GetProperties(  
    AcArray<AcMapObjClassProperty*>& aProperties,  
    AcArray<VARIANT>* paValues,  
    const AcDbEntity * pEntity  
) const;
```

Parameters	Description
aProperties	Output array of <a href="#">AcMapObjClassProperty</a> properties. The caller must free this object.
paValues	Output array of property values, or NULL if not needed by the caller. The array index of each property value matches the index of each corresponding property in aProperties. The name, category, and type properties echo the entity's current state. The default value is set to the property's current value. The read-only property is set to false. The visibility property is set to true. The range property is not set. The caller must free this object.
pEntity	Input entity to examine.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

Use this function to find all properties of a specific entity and decide which one to classify in the corresponding feature class definition. If you are not using the properties to update a feature class definition, this function can run without classification requirements (such as a feature-definition file or class name).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetProperties Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the properties and their values of an entity.

```
AcMapObjClass::EErrCode GetProperties(  
    AcArray<AcMapObjClassProperty*>& aProperties,  
    AcArray<VARIANT>* paValues,  
    const AcDbObjectId& entId  
) const;
```

Parameters	Description
aProperties	Output array of <a href="#">AcMapObjClassProperty</a> properties. The caller must free this object.
paValues	Output array of property values, or NULL if not needed by the caller. The array index of each property value matches the index of each corresponding property in aProperties. The name, category, and type properties echo the entity's current state. The default value is set to the property's current value. The read-only property is set to false. The visibility property is set to true. The range property is not set. The caller must free this object.
entId	Input ID of the entity to examine.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

Use this function to find all properties of a specific entity and decide which one to classify in the corresponding feature class definition. If you are not using the properties to update a feature class definition, this function can run without classification requirements (such as a feature-definition file or class name).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager:: GetUnclassifiedEntities Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the unclassified entities in the current drawing.

```
bool GetUnclassifiedEntities(  
    AcDbObjectIdArray& aEntIds  
) const;
```

Parameters

Description

aEntIds

Output IDs of all unclassified entities.

Returns

Returns true if at least one entity ID is returned; otherwise, returns false if no entity ID is returned or the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: GetUndefinedEntities Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Lists all the undefined entities in the current drawing.

```
bool GetUndefinedEntities(  
    AcDbObjectIdArray& aEntIds  
) const;
```

Parameters	Description
aEntIds	Output IDs of all undefined entities.

Returns

Returns true if at least one entity ID is returned; otherwise, returns false if no entity ID is returned or the process failed.

## Remarks

A classified entity is undefined if it lacks a corresponding feature class definition in the attached feature-definition file, or if it was classified with a different feature-definition file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager::ReloadCurrentFeatureDefinitionFile Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Reloads the feature-definition file attached to the current drawing.

[AcMapObjClass::EErrCode](#) ReloadCurrentFeatureDefinitionFile();

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

Use this function with caution. The feature-definition file's data and file name are cached in memory, so if a user is modifying the file while this function executes, it may reload outdated data while the file and memory are unsynchronized (prior to the user saving the file). In particular, this function will fail if the user renames or deletes the feature-definition file.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
AcMapClassificationManager:: SaveCurrentFeatureDefinitionFile Method  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Saves the feature-definition file attached to the current drawing.

[AcMapObjClass::EErrCode](#) SaveCurrentFeatureDefinitionFile() **const**;  
Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eNoSchemaFileAttached if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) eFailedSavingSchema if the feature-definition file could not be saved. Returns [AcMapObjClass::EErrCode](#) eNoUserPrivilegeToAlterSchema if the current user lacks the privileges to change the feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)  
[AcMapClassificationManager::Unclassify Method](#)  
[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Unclassifies an entity. Unclassifying entities does not trigger internal transactions. Unclassify only entities classified with the currently attached feature-definition file. If you unclassify an entity that was classified with a different feature-definition file, the corresponding tags will not be changed. To remove all existing tags, regardless of which feature-definition file was used, use `ClearAllTags()`.

```
AcMapObjClass::EErrCode Unclassify(  
    AcDbObjectId& entId  
);
```

Parameters	Description
entId	Input ID of the entity to unclassify.

### Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if successful. Returns [AcMapObjClass::EErrCode](#) `eNoSchemaFileAttached` if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) `eAlreadyUnclassified` if the entity is already-unclassified. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapClassificationManager Class](#), [AcMapClassificationManager Class](#)

[AcMapClassificationManager::Unclassify Method](#)

[AcMapClassificationManager Class](#) | [AcMapClassificationManager Class](#)

Unclassifies multiple entities. Unclassifying entities does not trigger internal transactions. Unclassify only entities classified with the currently attached feature-definition file. If you unclassify an entity that was classified with a different feature-definition file, the corresponding tags will not be changed. To remove all existing tags, regardless of which feature-definition file was used, use `ClearAllTags()`.

```
AcMapObjClass::EErrCode Unclassify(  
    AcDbObjectIdArray* paAlreadyUnclassifiedIds,  
    AcDbObjectIdArray& aEntIds  
);
```

Parameters

Description

`paAlreadyUnclassifiedIds` Output IDs of already-unclassified entities, or NULL if not needed by the caller.

`aEntIds` Input IDs of the entities to unclassify.

Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if all entities are declassified successfully. Returns [AcMapObjClass::EErrCode](#) `eNoSchemaFileAttached` if no feature-definition file is attached to the current drawing. Returns [AcMapObjClass::EErrCode](#) `eFailed` if the process failed for at least one entity.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: ~AcMapObjClassDefinition Destructor  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Destroys an instance of this class.

```
virtual ~AcMapObjClassDefinition(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::AddProperty Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Adds a new property to this feature class definition.

```
AcMapObjClass::EErrCode AddProperty(  
    const AcMapObjClassProperty& property  
);
```

Parameters	Description
property	Input property to add.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) ePropertyAlreadyExists if the property already exists. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetFeatureDefinitionFile Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the full pathname of the feature-definition file that this feature class definition belongs to.

```
const ACHAR* GetFeatureDefinitionFile() const;
```

Returns

Returns the feature-definition file's full pathname.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetName Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Retrieves the name of this feature class definition.

```
const ACHAR* GetName() const;
```

Returns

Returns the class name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: GetProperties Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Lists all the classified properties of this feature class definition.

```
AcMapObjClass::EErrCode GetProperties(  
    AcArray<AcMapObjClassProperty*>& aProperties  
) const;
```

Parameters	Description
aProperties	Output list of <a href="#">AcMapObjClassProperty</a> classified properties. The caller must free this object.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) ePropertyNotFound if none classified. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: GetSupportedEntityTypes Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Lists the AutoCAD entity types that this feature class definition supports.

```
AcMapObjClass::EErrCode GetSupportedEntityTypes(  
    AcArray<AcRxClass*>& aEntityTypes,  
    AcMapStringArray& aBlockNames  
) const;
```

Parameters	Description
aEntityTypes	Output array of supported entity types.
aBlockNames	Output array of supported block names.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if the class supports at least one entity type. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eRxClassNotFound if an entity type was found but the corresponding AcRxClass was not found in the class dictionary. Returns [AcMapObjClass::EErrCode](#) eFailed if no entity type is supported or if the process failed for some other reason.

## Remarks

The list typically matches the value that AcRxObject::desc() returns. A feature class definition can support any block reference, or specific block reference names. For any block reference, the corresponding entity type AcDbBlockReference::desc() is output in aEntityTypes. For specific block reference names, the names are output in aBlockNames.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: GetSupportedEntityTypes Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Lists the AutoCAD entity types that this feature class definition supports.

```
AcMapObjClass::EErrCode GetSupportedEntityTypes(  
    AcMapStringArray& aEntityTypes,  
    AcMapStringArray& aBlockNames  
) const;
```

Parameters	Description
aEntityTypes	Output array of supported entity types.
aBlockNames	Output array of supported block names.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if the class supports at least one entity type. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if no entity type is supported or if the process failed for some other reason.

Remarks

The list typically matches the value that AcRxClass::name() returns. A feature class definition can support any block reference, or specific block reference names. For any block reference, the corresponding entity type AcDbBlockReference::desc()->name() is output in aEntityTypes. For specific block reference names, the names are output in aBlockNames.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsBaseClassOnly Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether this feature class definition is a strict base class.

```
AcMapObjClass::EErrCode IsBaseClassOnly(  
    bool* pbIsBaseOnly  
) const;
```

Parameters	Description
pbIsBaseOnly	Output true if the feature class definition is a strict base class; otherwise, false.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

You can derive other classes from a strict base class but not classify objects with it.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition::IsVisibleInWorkspace Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Determines whether this feature class definition is visible in the AutoCAD Map project workspace.

```
AcMapObjClass::EErrCode IsVisibleInWorkspace(  
    bool* pbIsVisible  
) const;
```

Parameters

Description

pbIsVisible

Output true if the feature class definition is visible, or false if it is invisible.

Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)

AcMapObjClassDefinition:: LinkedDataMovedToFdo Method

[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

This function allows a client program to find out whether a link to the external data is going to be preserved to the FDO or the whole record is conveyed to the FDO.

[AcMapObjClass::EErrCode](#) LinkedDataMovedToFdo(  
    **bool** & bDataMove

) **const**;

Parameters

Description

bDataMove

Output value is true if the only link to external data preserved to FDO. Output is false if the actual data is preserved to the FDO.

Returns

Returns AcMapObjClass::eOk if successful. Returns AcMapObjClass::eClassNotFromCurrentSchema if class not from current schema. Returns AcMapObjClass::eFailed if fails for any reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: SetLinkedDataMovedToFdo Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

This function allows a client program to define the behavior of the linked data in process of querying and saving from/to FDO. A link to the external data only will be preserved to the FDO if the boolean parameter is set to true. Otherwise, records will be moved to the FDO. conveyed to the FDO.

```
AcMapObjClass::EErrorCode SetLinkedDataMovedToFdo(  
    bool bDataMove  
);
```

Parameters	Description
bDataMove	Input value. True if only link to external data preserved to FDO. False if the actual data is preserved to the FDO.

## Returns

Returns AcMapObjClass::eOk if successful. Returns AcMapObjClass::eClassNotFromCurrentSchema if class not from current schema. Returns AcMapObjClass::eFailed if fails for any reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassDefinition Class](#), [AcMapObjClassDefinition Class](#)  
AcMapObjClassDefinition:: SetVisibleInWorkspace Method  
[AcMapObjClassDefinition Class](#) | [AcMapObjClassDefinition Class](#)

Sets this feature class definition to visible or invisible in the AutoCAD Map project workspace.

```
AcMapObjClass::EErrCode SetVisibleInWorkspace(  
    bool bVisible  
);
```

Parameters	Description
bVisible	Input true for visible or false for invisible.
Returns	

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) eClassNotFromCurrentSchema if the class is not from the current feature-definition file. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

A visible class will not appear in the project workspace if one of its parent classes is invisible.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)  
AcMapObjClassProperty:: ~AcMapObjClassProperty Destructor  
[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Destroys an instance of this class.

```
virtual ~AcMapObjClassProperty();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: GetCategory Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the direct category of this property.

```
void GetCategory(  
    AcMapStringArray& aCategories  
) const;
```

Parameters	Description
aCategories	Output category.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)  
AcMapObjClassProperty:: GetDefaultValue Method  
[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the default value of this property.

```
bool GetDefaultValue(  
    VARIANT * pvarDefault  
) const;
```

Parameters	Description
pvarDefault	Output default value. The caller must free this object, typically with VariantClear().

## Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty::GetName Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the name of this property.

```
const ACHAR* GetName() const;
```

Returns

Returns the property's name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: GetRange Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the range of valid values for this property.

```
const ACHAR* GetRange() const;
```

Returns

Returns the property's range of valid values, or "--" if the range has not been set with SetRange().

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: GetType Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the type of this property.

VARTYPE GetType() **const**;

Returns

Returns the property's type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: GetValue Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Retrieves the value of this property from the specified entity.

```
AcMapObjClass::EErrCode GetValue(  
    VARIANT * pvarValue,  
    const AcDbObjectId& entId  
) const;
```

Parameters	Description
pvarValue	Output value of the property.
entId	Input object ID of the entity whose property value to get.

## Returns

Returns [AcMapObjClass::EErrCode](#) eOk if successful. Returns [AcMapObjClass::EErrCode](#) ePropertyNotFound if the property is not present on the entity. Returns [AcMapObjClass::EErrCode](#) eFailed if the process failed for some other reason.

## Remarks

The entity and property can be classified or unclassified.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: IsInRange Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Determines whether a value falls in this property's range of valid values.

```
bool IsInRange(  
    VARIANT * pVar  
);
```

Parameters	Description
pVar	Input value to examine.

Returns

Returns true if the value is in the range, or false if it is out of range or the range has not been set with SetRange().

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: IsReadOnly Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Determines whether this property is read-only or read-write.

```
bool IsReadOnly() const;
```

Returns

Returns true if the property is read-only, or false if it is read-write.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

AcMapObjClassProperty:: isVisible Method

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Determines whether this property is visible or invisible.

```
bool isVisible() const;
```

Returns

Returns true if the property is visible, or false if it is invisible.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: SetDefaultValue Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Sets the default value of this property.

```
AcMapObjClass::EErrorCode SetDefaultValue(  
    VARIANT * pvarDefault  
);
```

Parameters	Description
pvarDefault	Input default value.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eInvalidType if the type is invalid. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the property's class is not in the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eOutOfRange if the value falls outside the valid range. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: SetReadOnly Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Sets this property to read-only or read-write.

```
AcMapObjClass::EErrorCode SetReadOnly(  
    bool bReadOnly  
);
```

Parameters

Description

bReadOnly

Input true to set the property to read-only, or false to set it to read-write.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the property's class is not in the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty::SetValue Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Sets the value of this property for the specified entity. The entity and property can be classified or unclassified. If this property is classified, this function checks the property's range of valid values and, if the value is invalid, sets the value to its default value if `bFixOutOfRangeIfClassified` is true. Properties retrieved with [AcMapClassificationManager::GetClassifiedProperties\(\)](#) (twoforms) are classified properties. Properties retrieved with [AcMapClassificationManager::GetProperties\(\)](#) (twoforms) are unclassified properties, and `SetValue()` sets property values regardless of the value of `bFixOutOfRangeIfClassified`.

```
AcMapObjClass::EErrCode SetValue(  
    const AcDbObjectId& entId,  
    const VARIANT * pvarValue,  
    bool bFixOutOfRangeIfClassified  
);
```

Parameters	Description
<code>entId</code>	Input object ID of the entity whose property value to set.
<code>pvarValue</code>	Input value to set the property to.
<code>bFixOutOfRangeIfClassified</code>	Input true to reset an out-of-range classified-property value to its default value, or false not to reset it.

## Returns

Returns [AcMapObjClass::EErrCode](#) `eOk` if successful. Returns [AcMapObjClass::EErrCode](#) `ePropertyNotFound` if the entity does not have the specified property. Returns [AcMapObjClass::EErrCode](#) `ePropertyReadOnly` if the property is a read-only property, or if the property is classified and [AcMapObjClassProperty::IsReadOnly\(\)](#) is true. Returns [AcMapObjClass::EErrCode](#) `eOutOfRange` if the property value is still out of range. This error code is returned only if `bFixOutOfRangeIfClassified` is true, the property is classified, and the out-of-range property value could not be reset. For example, if the range of the Layer property specifies a layer that does not

exist in the current drawing, an attempt to fix this nonexistent layer will fail. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

AcMapObjClassProperty:: SetVisible Method

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Sets this property to visible or invisible.

```
AcMapObjClass::EErrorCode SetVisible(  
    bool bVisible  
);
```

Parameters

Description

bVisible

Input true to set the property to visible, or false to set it to invisible.

Returns

Returns [AcMapObjClass::EErrorCode](#) eOk if successful. Returns [AcMapObjClass::EErrorCode](#) eClassNotFromCurrentSchema if the property's class is not in the current feature-definition file. Returns [AcMapObjClass::EErrorCode](#) eFailed if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassProperty Class](#), [AcMapObjClassProperty Class](#)

[AcMapObjClassProperty:: ToString Method](#)

[AcMapObjClassProperty Class](#) | [AcMapObjClassProperty Class](#)

Returns the string representation of a value.

```
ACHAR* ToString(  
    const VARIANT* pvarValue  
) const;
```

Parameters	Description
pvarValue	Input value.

Returns

Returns the string value, or NULL if the type is invalid. The caller must free this object, typically with `acutDelString()`.

## Remarks

The value's type must be the same as that of the property.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)  
[AcMapObjClassReactor:: ~AcMapObjClassReactor Destructor](#)  
[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Destroys an instance of this class.

```
virtual ~AcMapObjClassReactor(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassReactor Class](#), [AcMapObjClassReactor Class](#)  
AcMapObjClassReactor:: AcMapObjClassReactor Constructor  
[AcMapObjClassReactor Class](#) | [AcMapObjClassReactor Class](#)

Constructs an instance of this class.

```
AcMapObjClassReactor(  
    void
```

```
);
```

Parameters	Description
------------	-------------

void	Void.
------	-------

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassSystem Class](#), [AcMapObjClassSystem Class](#)  
AcMapObjClassSystem:: RemoveObjClassReactor Method  
[AcMapObjClassSystem Class](#) | [AcMapObjClassSystem Class](#)

Removes a reactor from the list of the classification reactors.

```
bool RemoveObjClassReactor(  
    AcMapObjClassReactor* pReactor  
);
```

Parameters	Description
pReactor	Input <a href="#">AcMapObjClassReactor</a> reactor.

Returns

Returns true if successful. Returns false if the reactor was not found in the list of reactors.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassSystem Class](#), [AcMapObjClassSystem Class](#)  
[AcMapObjClassSystem::~~AcMapObjClassSystem Destructor](#)  
[AcMapObjClassSystem Class](#) | [AcMapObjClassSystem Class](#)

Destroys an instance of this class.

```
virtual ~AcMapObjClassSystem(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapObjClassSystem Class](#), [AcMapObjClassSystem Class](#)  
AcMapObjClassSystem:: AcMapObjClassSystem Constructor  
[AcMapObjClassSystem Class](#) | [AcMapObjClassSystem Class](#)

Constructs an instance of this class.

```
AcMapObjClassSystem(  
    void
```

```
);
```

Parameters	Description
------------	-------------

void	Void.
------	-------

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapObjClassSystem Class](#), [AcMapObjClassSystem Class](#)

AcMapObjClassSystem:: AddObjClassReactor Method

[AcMapObjClassSystem Class](#) | [AcMapObjClassSystem Class](#)

Adds a reactor to the list of the classification reactors.

```
bool AddObjClassReactor(  
    AcMapObjClassReactor* pReactor  
);
```

Parameters

Description

pReactor

Input [AcMapObjClassReactor](#)reactor.

Returns

Returns true if successful. Returns false if the reactor is already active.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#)

AcMapGeometryPE:: EGeometryType Enumeration

[AcMapGeometryPE Class](#)

Types for simple geometry elements.

```
enum EGeometryType {  
    kUnsupported = 0,  
    kPoint = 1,  
    kPolyline = 2,  
    kPolygon = 3  
};
```

File

AcMapGeometryPE.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE:: beginRead Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Executes before reading an entity begins.

```
virtual void beginRead(  
    AcDbEntity * pEnt  
);
```

Parameters	Description
------------	-------------

pEnt	The entity.
------	-------------

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE:: endRead Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Executes after reading an entity ends.

```
virtual void endRead(  
    AcDbEntity * pEnt  
);
```

Parameters	Description
------------	-------------

pEnt	The entity.
------	-------------

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE:: getGeometryType Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Gets the type of a simple element.

```
virtual AcMapGeometryPE::EGeometryType getGeometryType(  
    AcDbEntity * pEnt,  
    unsigned int geomIndex  
);
```

Parameters	Description
pEnt	The entity containing the element.
geomIndex	The element index.

Returns

Returns the type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE::getNumGeometries Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Counts simple elements in an entity.

```
virtual unsigned int getNumGeometries(  
    AcDbEntity * pEnt  
);
```

Parameters	Description
------------	-------------

pEnt	The entity.
------	-------------

Returns

Returns the count.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE:: getNumVertices Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Counts vertices of a polyline or polygon.

```
virtual unsigned int getNumVertices(  
    AcDbEntity * pEnt,  
    unsigned int geomIndex  
);
```

Parameters	Description
pEnt	The entity containing the element.
geomIndex	The element index.

Returns

Returns the count.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE:: getPoint Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Gets a point element.

```
virtual Acad::ErrorStatus getPoint(  
    AcGePoint3d & p,  
    AcDbEntity * pEnt,  
    unsigned int geomIndex  
);
```

Parameters	Description
p	Output point in WCS coordinates.
pEnt	The entity containing the element.
geomIndex	The element index.

Returns

Returns Acad::eOk on success.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapGeometryPE Class](#), [AcMapGeometryPE Class](#)

[AcMapGeometryPE::getVertex Method](#)

[AcMapGeometryPE Class](#) | [AcMapGeometryPE Class](#)

Gets a vertex of a polyline or polygon.

```
virtual Acad::ErrorStatus getVertex(  
    AcGePoint3d & pt,  
    double & bulge,  
    AcGeVector3d & normal,  
    AcDbEntity * pEnt,  
    unsigned int geomIndex,  
    unsigned int vertexIndex  
);
```

Parameters	Description
pt	Output point in WCS coordinates.
bulge	Output segment bulge.
normal	Output segment normal.
pEnt	Input the entity containing the element.
geomIndex	Input the element index.
vertexIndex	Input the vertex index.

Returns

Returns Acad::eOk on success.

Remarks

A segment bulge greater than zero indicates that the next segment is an arc. Where the next segment is an arc, segment normal identifies the plane in which the arc lies. Note that segment bulge at the last vertex is not defined, and neither is segment normal, because there is no next segment.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#)

AcMapQueryPE:: ELocationType Enumeration

[AcMapQueryPE Class](#)

Location-query process types.

```
enum ELocationType {  
    kBoundingBox = 1,  
    kIntersectWith = 2,  
    kAbstractGeometry = 3  
};  
File
```

AcMapQueryPE.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#)

AcMapQueryPE:: EPropType Enumeration

[AcMapQueryPE Class](#)

Property types for AutoCAD Map entities.

```
enum EPropType {  
    kElevation = 1,  
    kThickness = 2,  
    kBlockId = 3,  
    kTextStyleId = 4,  
    kTextString = 5,  
    kArea = 6,  
    kLength = 7,  
    kRotation = 8,  
    kHeight = 9,  
    kScale = 10,  
    kWidth = 11,  
    kBulge = 12,  
    kStartAngle = 13,  
    kEndAngle = 14,  
    kStartWidth = 15,  
    kEndWidth = 16,  
    kRadius = 17,  
    kShapeName = 18,  
    kShapeSize = 19,  
    kImageName = 20,  
    kAttributeTag = 21,  
    kPoint1 = 22,  
    kPoint2 = 23,  
    kPoint3 = 24,  
    kPoint4 = 25  
};
```

File

AcMapQueryPE.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE:: canSaveBack Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Can the entity be saved back?.

```
virtual bool canSaveBack(  
    AcDbEntity * pEnt = NULL,  
    AcDbDatabase * toDb = NULL  
);
```

Parameters	Description
pEnt	The entity.
toDb	Database to save back to.

## Returns

Returns true if the entity can be saved back.

## Remarks

entity can be NULL, meaning you are being asked about all entities of this type.  
database will be also NULL in this case.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE:: getPointsOnEntity Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Get any physical point on the entity to check whether it is inside location boundary or outside.

```
virtual Acad::ErrorStatus getPointsOnEntity(  
    AcGePoint3dArray & pts,  
    AcDbEntity * pEnt  
);
```

Parameters	Description
pts	Output a set of points.
pEnt	The entity.

Returns

Returns Acad::eOk on success.

## Remarks

In case of complex geometry (like multi-point or multi-line) the function should return multiple points, one point per subgeometry. Default getStretchPoint() rarely gives the right result.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE::getPropertyValue Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Gets a property value of an entity.

```
virtual Acad::ErrorStatus getPropertyValue(  
    AcMapValue & val,  
    AcDbEntity * pEnt,  
    AcMapQueryPE::EPropType type  
);
```

Parameters	Description
val	Output property value.
pEnt	The entity.
type	Property type.

Returns

Returns Acad::eOk on success.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE:: hasProperty Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Does the entity have a given property?.

```
virtual bool hasProperty(  
    AcDbEntity * pEnt,  
    AcMapQueryPE::EPropType type  
);
```

Parameters	Description
pEnt	The entity.
type	Property type.

Returns

Returns true if it has the property.

## Remarks

These are non-AcDbEntity properties used in property queries, property alteration, and AutoCAD Map expressions.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE:: isClosed Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

One more property alteration functions telling whether we need to apply hatch alteration to the entity or not.

```
virtual Acad::ErrorStatus isClosed(  
    bool & bClosed,  
    AcDbEntity * pEnt  
);
```

Parameters	Description
bClosed	Return flag.
pEnt	The entity.
Returns	

Returns Acad::eOk on success.

## Remarks

If not implemented, the default behaviour is to hatch MPolygons and closed AcDbCurves.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE:: isPropertyReadOnly Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Can an entity be changed by property alteration?.

```
virtual bool isPropertyReadOnly(  
    AcDbEntity * pEnt,  
    AcMapQueryPE::EPropType type  
);
```

Parameters	Description
pEnt	The entity.
type	Property type to alter.

Returns

Returns true if the property can be altered.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE::locationQueryType Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Specifies how the entity will be processed in a location query.

```
virtual AcMapQueryPE::ELocationType locationQueryType(  
    AcDbEntity * pEnt  
);
```

Parameters	Description
pEnt	The entity.

Returns

Returns a location-query process type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE:: onClosedSet Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Sends a notification to the entity at the time of closed set calculation (query and save back).

```
virtual void onClosedSet(  
    AcDbObjectIdArray & ids,  
    AcDbEntity * pEnt  
);
```

Parameters	Description
ids	Default closed set, which can be changed inside this function.
pEnt	The entity.
Returns	

Returns nothing.

Remarks

Default implementation is `AcDbObject::dwgOut()` with `AcDb::kIdFiler`.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)  
AcMapQueryPE:: setPropertyValue Method  
[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Performs a property alteration on an entity.

```
virtual Acad::ErrorStatus setPropertyValue(  
    AcDbEntity * pEnt,  
    AcMapQueryPE::EPropType type,  
    const AcMapValue & val  
);
```

Parameters	Description
pEnt	The entity.
type	Property type.
val	Value to set.

Returns

Returns Acad::eOk on success.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE::swapIdWith Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Swap ID with the prototype entity during saving back.

```
virtual Acad::ErrorStatus swapIdWith(  
    AcDbEntity * pEnt,  
    AcDbObjectId otherId,  
    Adesk::Boolean swapXdata,  
    Adesk::Boolean swapExtDict  
);
```

Parameters	Description
pEnt	The entity.
otherId	ID of the prototype entity.
swapXdata	Input true to swap extended entity data.
swapExtDictionary	Input true to swap extension dictionary.

Returns

Returns Acad::eOk on success.

Remarks

Saving back involves the following steps:

- Clone objects to the original database using wblockCloneObjects().
- Swap Ids for all old and new copies using AcDbObject::swapIdWith().
- Swap all references using AcDbObject::swapReferences().
- Erase old copies.

There are two virtual AcDbObject methods which could be used by custom objects to participate in the process: subSwapIdWith() and swapReferences(). The swapIdWith method lets you control swapping Xdata and Extension dictionaries. By default, they are not swapped.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapQueryPE Class](#), [AcMapQueryPE Class](#)

[AcMapQueryPE::transformBy Method](#)

[AcMapQueryPE Class](#) | [AcMapQueryPE Class](#)

Transforms the entity (projection transformation during query and save back, and rubber-sheeting).

```
virtual Acad::ErrorStatus transformBy(  
    AcDbEntity * pEnt,  
    bool bMirror,  
    bool bSimple,  
    Acad::ErrorStatus (*getMatrixInPoint) (AcGeMatrix3d &mat, bool b  
);
```

### Parameters

### Description

pEnt

The entity.

bSimple

Output true if transformation matrix is constant, and you can call regular transformBy().

bMirror

Output true if you are rubber-sheeting and you need to swap all angles and bulges.

getMatrixInPoint

Callback function to calculate transformation matrix at the specified point.

mat

Output the result matrix.

bUniScaleOrtho

Output true if you need the matrix to be isUniScaledOrtho() because rubber-sheeting will create a non-orthogonal matrix.

p

The point to calculate matrix at.

### Returns

Returns Acad::eOk on success.

### Remarks

This is non-linear transformation, where matrix is different in the different points. Default implementation calculates a constant matrix at some middle point on the entity, and calls AcDbEntity::transformBy() with that constant matrix.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)  
AcMapDataSources:: ~AcMapDataSources Destructor  
[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Destroys an instance of this class.

```
~AcMapDataSources(  
    void  
);  
Returns
```

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)  
AcMapDataSources:: AcMapDataSources Constructor  
[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Constructs an instance of this class.

```
AcMapDataSources(  
    void  
);  
Returns
```

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)  
AcMapDataSources:: DetachAllDataSources Method  
[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Detaches all data sources from the current AutoCAD Map work session.

```
void DetachAllDataSources(  
    void  
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

AcMapDataSources:: DisconnectAllDataSources Method

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Disconnects all data sources from the current AutoCAD Map work session.

```
void DisconnectAllDataSources(  
    void  
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

[AcMapDataSources:: GetAttachedDataSources Method](#)

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Retrieves the names of the attached data sources in the current AutoCAD Map work session.

```
void GetAttachedDataSources(  
    AcMapStringArray& paDataSourceNames  
);
```

Parameters	Description
paDataSourceNames	Output array of the names of the attached data sources.

## Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

AcMapDataSources:: GetAttachedDataSourcesCount Method

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Counts the number of attached data sources in the current AutoCAD Map work session.

```
long GetAttachedDataSourcesCount(  
    void  
);
```

Returns

Returns the number of attached data sources.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

[AcMapDataSources:: GetConnectedDataSources Method](#)

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Retrieves the names of the connected data sources in the current AutoCAD Map work session.

```
void GetConnectedDataSources(  
    AcMapStringArray& paDataSourceNames  
);
```

Parameters	Description
paDataSourceNames	Output array of the names of the connected data sources.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

AcMapDataSources:: GetConnectedDataSourcesCount Method

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Counts the number of connected data sources in the current AutoCAD Map work session.

```
long GetConnectedDataSourcesCount(  
    void  
);
```

Returns

Returns the number of connected data sources.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDataSources Class](#), [AcMapDataSources Class](#)

[AcMapDataSources:: GetDisconnectedDataSources Method](#)

[AcMapDataSources Class](#) | [AcMapDataSources Class](#)

Retrieves the names of the disconnected data sources in the current AutoCAD Map work session.

```
void GetDisconnectedDataSources(  
    AcMapStringArray& paDataSourceNames  
);
```

Parameters	Description
paDataSourceNames	Output array of the names of the disconnected data sources.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMAddScaleFactorReactor Function

[AcMapDMScaleFactor Namespace](#)

Adds a scale-factor reactor to monitor scale-factor changes.

```
void DMAddScaleFactorReactor(  
    DMScaleFactorReactor * pReactor  
);  
File
```

DmScaleFactor.h

Parameters	Description
pReactor	Input scale-factor reactor.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMDisplayIsMetric Function

[AcMapDMScaleFactor Namespace](#)

Determines the display measurement units.

```
bool DMDisplayIsMetric();
```

File

DmScaleFactor.h

Returns

Returns true for centimeters, or false for inches.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMDwgUnitsAreMetric Function

[AcMapDMScaleFactor Namespace](#)

Determines the dwg measurement units.

```
bool DMDwgUnitsAreMetric(  
    double& dDwgConversionFactor,  
    AcDbDatabase* pDb  
);  
File
```

DmScaleFactor.h

Parameters	Description
dDwgConversionFactor	Output double for converting from dwg units to inches or centimeters
pDb	Input AcDbDatabase to test for metric dwg units.
Returns	

Returns true for metric, or false for imperial.

Remarks

Intended for internal use only.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMGetCurrentScaleFactor Function

[AcMapDMScaleFactor Namespace](#)

Retrieves the cached current scale factor.

```
double DMGetCurrentScaleFactor();
```

File

DmScaleFactor.h

Returns

Returns the cached current scale factor.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor:: DMRefreshScale Function

[AcMapDMScaleFactor Namespace](#)

Forces the scale factor to update.

```
void DMRefreshScale();
```

File

DmScaleFactor.h

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

AcMapDMScaleFactor::DMRemoveScaleFactorReactor Function

[AcMapDMScaleFactor Namespace](#)

Removes a scale-factor reactor.

```
void DMRemoveScaleFactorReactor(  
    DMScaleFactorReactor * pReactor  
);  
File
```

DmScaleFactor.h

Parameters	Description
pReactor	Input scale-factor reactor.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMScaleFactor Namespace](#)

DMScaleFactorReactor Class

[AcMapDMScaleFactor Namespace](#)

Reactor class for scale-factor changes.

```
class DMScaleFactorReactor;
```

File

DmScaleFactor.h

☐ Methods



[~DMScaleFactorReactor](#) Destroys an instance of this class.



[DMScaleFactorReactor](#) Constructs an instance of this class.

[ScaleFactorChanged](#) Invoked when the scale factor changes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: EDMStatus Enumeration

[AcMapDMDisplayManagement Namespace](#)

Enumerates the types of display-management active status.

```
enum EDMStatus {  
    kProcessing = 0x0001  
};  
File
```

DmDisplayManagement.h

Parameters	Description
kPassive	Passive.
kActive	Active.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: AddProjectReactor Function

[AcMapDMDisplayManagement Namespace](#)

Adds a display-management reactor to monitor project-level activities such as creation and deletion.

```
bool AddProjectReactor(  
    AcMapProject * pProject,  
    AcMapDMProjectReactor* pDMReactor  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pProject	Input AutoCAD Map project.
pDMReactor	Input <a href="#">AcMapDMProjectReactor</a> .
Returns	

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: CommitEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to commit uncommitted edits to objects the Element queried in from external data sources.

```
Acad::ErrorStatus CommitEdits(  
    AcMapDMElement* pElement  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pElement	Input Element for which this call applies.
Returns	

Returns Acad::eOk if successful. Returns Acad::eNotApplicable if pElement->ClonesObjectsFromExternalSource() is false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: CommitEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to commit uncommitted edits to objects the Elements within the specified Map queried in from external data sources.

```
Acad::ErrorStatus CommitEdits(  
    AcMapDMMap* pMap  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pMap	Input Map for which this call applies.
Returns	

Returns Acad::eOk if successful. Returns Acad::eNotApplicable if all of the Map's Elements'

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DiscardEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to discard uncommitted edits to objects the Element queried in from external data sources.

```
Acad::ErrorStatus DiscardEdits(  
    AcMapDMElement* pElement  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pElement	Input Element for which this call applies.
Returns	

Returns Acad::eOk if successful. Returns Acad::eNotApplicable if pElement->ClonesObjectsFromExternalSource() is false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DiscardEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to discard uncommitted edits to objects the Elements within the specified Map queried in from external data sources.

```
Acad::ErrorStatus DiscardEdits(  
    AcMapDMMap* pMap  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pMap	Input Map for which this call applies.
Returns	

Returns Acad::eOk if successful. Returns Acad::eNotApplicable if all of the Map's Elements'

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DMMMapManagerId Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the ID of the display-management map manager object for a database.

exist

```
AcDbObjectId DMMMapManagerId(  
    AcDbDatabase * pDatabase,  
    bool bCreate = true  
);
```

File

DmDisplayManagement.h

Parameters	Description
pDatabase	Input database.
bCreate	Input true to create a map manager if one does not already exist. The default value is true.

Returns

Returns the ID of the AcMapDMMapManager.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DMMapManagerId Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the ID of the display-management map manager object for a project.

```
AcDbObjectId DMMapManagerId(  
    AcMapProject * pProject,  
    bool bCreate = true  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pProject	Input AutoCAD Map project.
bCreate	Input true to create a map manager if one does not already exist. The default value is true.

Returns

Returns the ID of the AcMapDMMapManager.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DMStyleLibraryId Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the ID of display-management style library object for a database.

```
AcDbObjectId DMStyleLibraryId(  
    AcDbDatabase * pDb  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pDb	Input database.
Returns	

Returns the ID of the AcMapDMStyleLibrary.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: DMStyleLibraryId Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the ID of display-management style library object for a project.

```
AcDbObjectId DMStyleLibraryId(  
    AcMapProject * pProject  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pProject	Input AutoCAD Map project.
Returns	

Returns the ID of the AcMapDMStyleLibrary.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement::

GetCurrentMapBaseElementInvisibleEntities Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the visibility status of the map base element and the IDs of entities made invisible when it was unchecked.

```
bool GetCurrentMapBaseElementInvisibleEntities(  
    bool& bMapBaseElemInvisible,  
    AcDbObjectIdArray& objIds  
);  
File
```

DmDisplayManagement.h

Parameters	Description
bMapBaseElemInvisible	Output flag for map base visibility (true indicates invisible).
objIds	Output array of IDs of entities that were made invisible. If the map base element is visible, this array will have zero length.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetDMMapManager Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the display-management map manager object for the specified database.

```
Acad::ErrorStatus GetDMMapManager(  
    AcMapDMMapManager*& pMapMan,  
    AcDbDatabase * pDatabase,  
    AcDb::OpenMode mode  
);
```

File

DmDisplayManagement.h

Parameters	Description
pMapMan	Output AcMapDMMapManager.
pDatabase	Input database.
mode	Input mode to open the map manager in.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetDMMMapManager Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the display-management map manager object for a specified AutoCAD Map project.

```
Acad::ErrorStatus GetDMMMapManager(  
    AcMapDMMapManager*& pMapMan,  
    AcMapProject * pProject,  
    AcDb::OpenMode mode  
);
```

File

DmDisplayManagement.h

Parameters	Description
pMapMan	Output AcMapDMMapManager.
pProject	Input AutoCAD Map project.
mode	Input mode to open the map manager in.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

If mode is AcDb::kForWrite and the objects do not exist, an empty map manager is created.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetDMStyleLibrary Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the display-management style library object for a database.

```
Acad::ErrorStatus GetDMStyleLibrary(  
    AcMapDMStyleLibrary*& pStyleLib,  
    AcDbDatabase * pDb,  
    AcDb::OpenMode mode
```

```
);
```

File

DmDisplayManagement.h

Parameters	Description
pStyleLib	Output AcMapDMStyleLibrary.
pDb	Input database.
mode	Input mode to open the library in.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: GetDMStyleLibrary Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the display-management style library object for a project.

```
Acad::ErrorStatus GetDMStyleLibrary(  
    AcMapDMStyleLibrary*& pStyleLib,  
    AcMapProject * pProject,  
    AcDb::OpenMode mode
```

```
);
```

File

DmDisplayManagement.h

Parameters	Description
pStyleLib	Output AcMapDMStyleLibrary.
pProject	Input AutoCAD Map project.
mode	Input mode to open the library in.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: HasDMDData Function

[AcMapDMDisplayManagement Namespace](#)

Determines whether a project has display-management data.

```
bool HasDMDData(  
    AcMapProject * pProject  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pProject	Input AutoCAD Map project.
Returns	

Returns true if the project has display-management data.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: HasUncommittedEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to determine if an Element has uncommitted edits to objects it queried in from external data sources.

```
bool HasUncommittedEdits(  
    AcMapDMElement* pElement  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pElement	Input Element for which this call applies.
Returns	

Returns true if the Element has a dirty save set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: HasUncommittedEdits Function

[AcMapDMDisplayManagement Namespace](#)

Used to determine if a Map has uncommitted edits to objects queried in from external data sources by any of the Map's Elements.

```
bool HasUncommittedEdits(  
    AcMapDMMap* pMap  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pMap	Input Map for which this call applies.
Returns	

Returns true if any Element within the Map has a dirty save set.

Remarks

All Elements within the Map are checked.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: OnBeginWritingToDataStore Function

[AcMapDMDisplayManagement Namespace](#)

Notifies the display manager that an application is about to write a drawing database to an external data store.

```
void OnBeginWritingToDataStore(  
    AcDbDatabase * pDb  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pDb	Input drawing database to be written to an external data store.

Returns

Returns nothing.

Remarks

Display Manager removes all effects, if any, of stylization applied to entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: OnEndWritingToDataStore Function

[AcMapDMDisplayManagement Namespace](#)

Notifies the display manager that an application is finished writing a drawing database to an external data store.

```
void OnEndWritingToDataStore(  
    AcDbDatabase * pDb  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pDb	Input drawing database that was written to an external data store.

Returns

Returns nothing.

Remarks

Display Manager restores the effects, if any, of stylization.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: RemoveProjectReactor Function

[AcMapDMDisplayManagement Namespace](#)

Removes a display-management reactor from a project.

```
bool RemoveProjectReactor(  
    AcMapProject * pProject,  
    AcMapDMProjectReactor* pDMReactor  
);  
File
```

DmDisplayManagement.h

Parameters	Description
pProject	Input AutoCAD Map project.
pDMReactor	Input <a href="#">AcMapDMProjectReactor</a> .
Returns	

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDisplayManagement Namespace](#)

AcMapDMDisplayManagement:: Status Function

[AcMapDMDisplayManagement Namespace](#)

Retrieves the display-management current active status.

```
unsigned int Status();
```

File

DmDisplayManagement.h

Returns

Returns the bitwise combination of status bits defined by EDMStatus.

Remarks

Display management is active when it perform stylization operations.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: BuildRangeTables Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Builds range tables.

**virtual** [AcMapDMThematicBuildRangesErrorCode](#) BuildRangeTables();

Returns

Returns [AcMapDMThematicBuildRangesErrorCode::keOk](#) if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: AddReactor Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Adds a reactor to this map.

```
bool AddReactor(  
    AcMapDMMMapReactor* pReactor  
);
```

Parameters	Description
pReactor	Input AcMapDMMMapReactor to add.
Returns	

Returns true if successful.

## Remarks

A reactor monitors map notifications.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: RemoveReactor Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Removes a reactor from this map.

```
bool RemoveReactor(  
    AcMapDMMMapReactor* pReactor  
);
```

Parameters	Description
pReactor	Input AcMapDMMMapReactor to remove.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAAllDrawOrderItemsIterator Class](#),

[AcMapDMAAllDrawOrderItemsIterator Class](#)

AcMapDMAAllDrawOrderItemsIterator::

~AcMapDMAAllDrawOrderItemsIterator Destructor

[AcMapDMAAllDrawOrderItemsIterator Class](#) |

[AcMapDMAAllDrawOrderItemsIterator Class](#)

Destroys an instance of this class.

**virtual** `~AcMapDMAAllDrawOrderItemsIterator()`;

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAAllDrawOrderItemsIterator Class](#),

[AcMapDMAAllDrawOrderItemsIterator Class](#)

AcMapDMAAllDrawOrderItemsIterator:: Done Method

[AcMapDMAAllDrawOrderItemsIterator Class](#) |

[AcMapDMAAllDrawOrderItemsIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() const = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAAllDrawOrderItemsIterator Class](#),

[AcMapDMAAllDrawOrderItemsIterator Class](#)

AcMapDMAAllDrawOrderItemsIterator:: Next Method

[AcMapDMAAllDrawOrderItemsIterator Class](#) |

[AcMapDMAAllDrawOrderItemsIterator Class](#)

Advances to the next element in the iteration.

```
virtual bool Next() = 0;
```

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAAllDrawOrderItemsIterator Class](#),

[AcMapDMAAllDrawOrderItemsIterator Class](#)

AcMapDMAAllDrawOrderItemsIterator:: ObjectId Method

[AcMapDMAAllDrawOrderItemsIterator Class](#) |

[AcMapDMAAllDrawOrderItemsIterator Class](#)

Retrieves the ID of the current object in the iteration.

```
virtual AcDbObjectId ObjectId() const = 0;
```

Returns

Returns a valid ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllItemsIterator Class](#), [AcMapDMAllItemsIterator Class](#)  
AcMapDMAllItemsIterator:: ~AcMapDMAllItemsIterator Destructor  
[AcMapDMAllItemsIterator Class](#) | [AcMapDMAllItemsIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMAllItemsIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllItemsIterator Class](#), [AcMapDMAllItemsIterator Class](#)

AcMapDMAllItemsIterator:: Done Method

[AcMapDMAllItemsIterator Class](#) | [AcMapDMAllItemsIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() const = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllItemsIterator Class](#), [AcMapDMAllItemsIterator Class](#)

[AcMapDMAllItemsIterator:: Next Method](#)

[AcMapDMAllItemsIterator Class](#) | [AcMapDMAllItemsIterator Class](#)

Advances to the next element in the iteration.

```
virtual bool Next() = 0;
```

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllItemsIterator Class](#), [AcMapDMAllItemsIterator Class](#)

AcMapDMAllItemsIterator:: ObjectId Method

[AcMapDMAllItemsIterator Class](#) | [AcMapDMAllItemsIterator Class](#)

Retrieves the ID of the current object in the iteration.

```
virtual AcDbObjectId ObjectId() const = 0;
```

Returns

Returns a valid ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: ~AcMapDMAllStyleReferencesIterator

Destructor

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMAllStyleReferencesIterator\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: Done Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() const = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: Next Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Advances to the next element in the iteration.

```
virtual bool Next() = 0;
```

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: ObjectId Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Retrieves the ID of the current object in the iteration.

```
virtual AcDbObjectId ObjectId() const = 0;
```

Returns

Returns a valid ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: Rewind Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: StyleId Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Retrieves the ID of the style that the current object points to.

```
virtual AcDbObjectId StyleId() const = 0;
```

Returns

Returns a valid ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAllStyleReferencesIterator Class](#),

[AcMapDMAllStyleReferencesIterator Class](#)

AcMapDMAllStyleReferencesIterator:: ThresholdScale Method

[AcMapDMAllStyleReferencesIterator Class](#) |

[AcMapDMAllStyleReferencesIterator Class](#)

Retrieves the scale threshold of the current object.

```
virtual double ThresholdScale() const = 0;
```

Returns

Returns the scale threshold.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor::

~AcMapDMAttachedDwgsQueryDataSourceDescriptor Destructor

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMAttachedDwgsQueryDataSourceDescriptor\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),  
[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)  
AcMapDMAttachedDwgsQueryDataSourceDescriptor::  
AcMapDMAttachedDwgsQueryDataSourceDescriptor Constructor  
[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |  
[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMAttachedDwgsQueryDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor::

GetDrawingList Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Lists the drawings used as the scope of the query.

```
virtual Acad::ErrorStatus GetDrawingList(  
    AcMapStringArray* pDwgList  
) const;
```

Parameters

Description

pDwgList

Output array of strings. Each array entry is a file path, given in the same sequence as in the project.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor:: GetQuery Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Retrieves the query definition.

```
virtual Acad::ErrorStatus GetQuery(  
    struct resbuf*& pRb  
) const;
```

Parameters

Description

pRb

Output resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor:: SetDrawingList Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Sets the list of drawings to use as the scope of the query.

```
virtual Acad::ErrorStatus SetDrawingList(  
    const AcMapStringArray* pDwgList  
);
```

Parameters

Description

pDwgList

Input array of strings. Each array entry is a file path, given in the same sequence as in the project.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#),

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

AcMapDMAttachedDwgsQueryDataSourceDescriptor:: SetQuery Method

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#) |

[AcMapDMAttachedDwgsQueryDataSourceDescriptor Class](#)

Sets the query definition.

```
virtual Acad::ErrorStatus SetQuery(  
    const struct resbuf* pRb  
);
```

Parameters

Description

pRb

Input resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement::

~AcMapDMAttachedDwgsQueryElement Destructor

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMAttachedDwgsQueryElement\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement::

AcMapDMAttachedDwgsQueryElement Constructor

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Constructs an instance of this class.

```
AcMapDMAttachedDwgsQueryElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: AcquireEntities Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

See also

[AcMapDMAttachedDwgsQueryDataSourceDescriptor::SetDrawingList\(\)](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement::

ClonesObjectsFromExternalSource Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement::DismissEntities Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Erases entities that are part of this query element.

```
virtual Acad::ErrorStatus DismissEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

See also AcquireEntities().

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: dwgInFields Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: dwgOutFields Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: dxfInFields Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: dxfOutFields Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: GetAcquisitionCriteria Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters

Description

pDataSourceDsc

Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: SetAcquisitionCriteria Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters

Description

pDataSourceDsc

Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMAttachedDwgsQueryElement Class](#),

[AcMapDMAttachedDwgsQueryElement Class](#)

AcMapDMAttachedDwgsQueryElement:: SetVisible Method

[AcMapDMAttachedDwgsQueryElement Class](#) |

[AcMapDMAttachedDwgsQueryElement Class](#)

Makes the entities that are part of this element visible or invisible.

```
virtual Acad::ErrorStatus SetVisible(  
    bool bNewVal,  
    double dScale  
);
```

Parameters

Description

bNewVal

Input true to make the entities visible, or false to make them invisible.

dScale

Input scale at which to make entities visible.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: ~AcMapDMCurrentDwgQueryElement  
Destructor

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMCurrentDwgQueryElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement::

AcMapDMCurrentDwgQueryElement Constructor

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Constructs an instance of this class.

```
AcMapDMCurrentDwgQueryElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: AcquireEntities Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Runs the query against the current drawing.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement::

ClonesObjectsFromExternalSource Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Returns false (because objects from only the current drawing are selected).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: dwgInFields Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: dwgOutFields Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: dxfInFields Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: dxfOutFields Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: GetAcquisitionCriteria Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters

Description

pDataSourceDsc

Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: OnMapProjectInitialized Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters

Description

pMapProject

Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMCurrentDwgQueryElement Class](#),

[AcMapDMCurrentDwgQueryElement Class](#)

AcMapDMCurrentDwgQueryElement:: SetAcquisitionCriteria Method

[AcMapDMCurrentDwgQueryElement Class](#) |

[AcMapDMCurrentDwgQueryElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters

Description

pDataSourceDsc

Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#)

AcMapDMDefaultStyle:: DefaultStyleType Enumeration

[AcMapDMDefaultStyle Class](#)

Enumerates the types of default styles.

```
enum DefaultStyleType {  
    kEntityFade = 0  
};  
File
```

DmDefaultStyle.h

Parameters	Description
kEntityFade	Fade style.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)  
[AcMapDMDefaultStyle:: ~AcMapDMDefaultStyle Destructor](#)  
[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMDefaultStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: AcMapDMDefaultStyle Constructor](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Constructs an instance of this class.

```
AcMapDMDefaultStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: Apply Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Applies the style to an entity.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to which the style is applied.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: ClearFade Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Clears the fade. See also [IsFadeCleared\(\)](#).

```
Acad::ErrorStatus ClearFade();
```

## Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: clone Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle::copyFrom Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

AcMapDMDefaultStyle:: dwgInFields Method

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

AcMapDMDefaultStyle:: dwgOutFields Method

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: dxfInFields Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: dxfOutFields Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)  
[AcMapDMDefaultStyle:: GetFade Method](#)  
[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Retrieves the fade property. See also SetFade().

```
Acad::ErrorStatus GetFade(  
    Adesk::UInt8& fade  
) const;
```

Parameters	Description
fade	Output fade value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: GetStyleType Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Retrieves the style type. This function is intended for use in future versions of AutoCAD Map, when styles other than `DefaultStyleType::kEntityFade` may be available. See also `SetStyleType()`.

[AcMapDMDefaultStyle::DefaultStyleType](#) `GetStyleType() const;`

Returns

Returns the `DefaultStyleType` style type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: IsFadeCleared Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Determines whether the fade is cleared. See also [ClearFade\(\)](#).

**bool** IsFadeCleared() **const**;

Returns

Returns true if the fade is cleared, or false if it is not.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

AcMapDMDefaultStyle:: SetFade Method

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Sets the fade property. In this release of AutoCAD Map, this function simply turns the target entity to ACI color 9 (light gray). Future releases may permit other fade colors. See also GetFade().

```
Acad::ErrorStatus SetFade(  
    Adesk::UInt8 fade  
);
```

Parameters	Description
fade	Input fade value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)

[AcMapDMDefaultStyle:: SetStyleType Method](#)

[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Sets the style type. This function is intended for use in future versions of AutoCAD Map, when styles other than `DefaultStyleType::kEntityFade` may be available. See also `GetStyleType()`.

```
Acad::ErrorStatus SetStyleType(  
    AcMapDMDefaultStyle::DefaultStyleType type  
);
```

Parameters	Description
type	Input <code>DefaultStyleType</code> style type.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMDefaultStyle Class](#), [AcMapDMDefaultStyle Class](#)  
[AcMapDMDefaultStyle:: UnApply Method](#)  
[AcMapDMDefaultStyle Class](#) | [AcMapDMDefaultStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#)

AcMapDMElement:: EPreviewEntityType Enumeration

[AcMapDMElement Class](#)

Enumerates the preview entity types.

```
enum EPreviewEntityType {
    kPreviewArc = 1,
    kPreviewLine,
    kPreviewDefault = kPreviewLine,
    kPreviewPoint,
    kPreviewPolyline,
    kPreviewPolygon,
    kPreviewText,
    kPreviewBlock,
    kPreviewCustom
};
File
```

DmDisplayElement.h

Parameters	Description
kPreviewArc	Arc.
kPreviewLine	Line.
kPreviewDefault	Default preview entity type (a kPreviewLine).
kPreviewPoint	Point.
kPreviewPolyline	Polyline.
kPreviewPolygon	Polygon.
kPreviewText	Text.
kPreviewBlock	Block.
kPreviewCustom	Not supported in this release of AutoCAD Map.

Remarks

These preview types determine the type that will be shown for the thumbnail preview of styles that are applied to the element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

[AcMapDMElement:: ~AcMapDMElement Destructor](#)

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: AcMapDMElement Constructor

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Constructs an instance of this class.

```
AcMapDMElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: AcquireEntities Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Runs the query and retrieves and stores the IDs of objects that meet the selection criteria.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: audit Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters

Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: ClonesObjectsFromExternalSource Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Clones objects that are from the external source at AcquireEntities() time.

**virtual bool** ClonesObjectsFromExternalSource() **const**;

Returns

Returns or true if objects from an external source were cloned successfully; returns false if entities were acquired from the current drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement::DismissEntities Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Clears the acquired entities that are stored in this element.

```
virtual Acad::ErrorStatus DismissEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

A derived function can perform extra cleanup.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement::DismissStylization Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Clears the current stylization.

```
virtual Acad::ErrorStatus DismissStylization(  
    bool bDismissEntities  
);
```

Parameters

Description

bDismissEntities

Input true to clear the acquired entities.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: dwgInFields Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: dwgOutFields Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: dxfInFields Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: dxfOutFields Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: EnableStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Enables the style reference's style.

```
virtual Acad::ErrorStatus EnableStyle(  
    AcDbObjectId styleRefId,  
    bool bNewEnableStatus  
);
```

Parameters	Description
styleRefId	Input ID of the style reference.
bNewEnableStatus	Input true to enable the style, or false to disable it.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: FilterOutStylizationEntities Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Filters out those entities from the selection that, during stylization, acquired stylization entities of another element.

```
void FilterOutStylizationEntities(  
    AcDbObjectIdArray& prunedSelection,  
    const AcDbObjectIdArray& selectedIds  
) const;
```

Parameters	Description
prunedSelection	Output array of IDs of the filtered objects.
selectedIds	Input array containing all IDs of acquired objects.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: GetAcquiredEntities Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the current selection (defined by calling AcquireEntities()).

```
virtual Acad::ErrorStatus GetAcquiredEntities(  
    AcDbObjectIdArray& Ids  
) const;
```

Parameters

Description

Ids

Output array of object IDs of the acquired objects.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)  
AcMapDMElement:: GetAcquisitionCriteria Method  
[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the acquisition criteria for this element.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters	Description
pDataSourceDsc	Output <a href="#">AcMapDMDDataSourceDescriptor</a> data-source descriptor.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: GetAllStyleReferencesIterator Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves an iterator over a collection of style references (for the corresponding style set) applied to this element.

```
AcMapDMAllStyleReferencesIterator* GetAllStyleReferencesIterator(  
    double dScale = 0.,  
    bool bForward = true,  
    bool bSkipProxies = false  
) const;
```

Parameters	Description
dScale	Input scale. The default value is zero, the current scale.
bForward	Input true for forward traversal.
bSkipProxies	Input true to skip over StyleReferences that refer to proxy Styles

Returns

Returns the [AcMapDMAllStyleReferencesIterator](#).

Remarks

The iterator will iterate all StyleReferences, and StyleId() may return an Id to an AcDbProxyObject, unless the bSkipProxies argument is passed as true.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: GetCustomPreviewEntityName Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the custom preview entity name. See also

[SetCustomPreviewEntityName\(\)](#).

```
virtual const ACHAR* GetCustomPreviewEntityName() const;
```

Returns

Returns NULL for all preview entity types except kPreviewBlock and kPreviewCustom. For kPreviewBlock, a block name is returned. For kPreviewCustom, a service name is returned - service names are not yet supported in AutoCAD Map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: GetPreviewBlockId Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the AcDbObjectId of the block used for preview, if preview is of type kPreviewBlock. See also SetPreviewBlockId().

```
AcDbObjectId GetPreviewBlockId() const;
```

Returns

Returns AcDbObjectId with kNull value for all preview entity types except kPreviewBlock

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)  
AcMapDMElement:: GetPreviewEntityType Method  
[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Retrieves the preview entity type of this element. See also [SetPreviewEntityType\(\)](#).

```
virtual EPreviewEntityType GetPreviewEntityType() const;
```

Returns

Returns the EPreviewEntityType thumbnail preview type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)  
AcMapDMElement:: SetPreviewEntityType Method  
[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the preview entity type. See also [GetPreviewEntityType\(\)](#).

```
virtual Acad::ErrorStatus SetPreviewEntityType(  
    EPreviewEntityType ePreviewType  
);
```

Parameters	Description
ePreviewType	Input EPreviewEntityTypepreview entity type.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: HasStyleReference Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines whether this element has a specified style reference.

```
bool HasStyleReference(  
    const AcDbObjectId& styleRefId,  
    double dScale  
);
```

Parameters	Description
styleRefId	Input ID of the style reference.
dScale	Input scale.

Returns

Returns true if this element has the specified style reference; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: IsStyleApplied Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines whether this element has a style associated with it at a specified scale.

```
bool IsStyleApplied(  
    const AcDbObjectId& styleId,  
    double dScale,  
    bool bIgnoreDisabled = true  
) const;
```

Parameters	Description
styleId	Input ID of the style.
dScale	Input scale.
bIgnoreDisabled	Input true to ignore styles that are disabled at the specified scale. The default value is true.

Returns

Returns true if the style is applied.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: IsStylized Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines whether this element is stylized.

```
bool IsStylized() const;
```

Returns

Returns true if this element is stylized.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

[AcMapDMElement:: IsUniqueStyleSet Method](#)

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines whether the style set on this element is unique. See also [MakeStyleSetUnique\(\)](#).

```
bool IsUniqueStyleSet(  
    double dScale = 0.  
) const;
```

Parameters

Description

dScale

Input scale. The default value is zero, the current scale.

Returns

Returns true if the style set is unique, or false if the style set is shared among multiple scales.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: isVisible Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Determines whether this element is visible at the specified scale. See also `setVisible()`.

```
virtual bool isVisible(  
    double dScale = 0.  
) const;
```

Parameters

Description

dScale

Input scale at which visibility is evaluated. The default value is zero, the current scale.

Returns

Returns true if the element is visible at the specified scale; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

[AcMapDMElement:: MakeStyleSetUnique Method](#)

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Makes unique the style set associated with this element. See also [IsUniqueStyleSet\(\)](#).

```
Acad::ErrorStatus MakeStyleSetUnique(  
    bool bIsUnique,  
    double dScale = 0.  
);
```

Parameters	Description
bIsUnique	Input true to copy the style set associated with the specified map scale and use this copy for the specified and lower scales. Input false to delete the style set associated with the specified map scale, making the specified scale refer to the style set associated with higher scales.
dScale	Input scale at which to make the style set unique. The default value is zero, the current scale.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: OnMapProjectInitialized Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: OnObjectAppended Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Invoked when an object is appended to the database.

**virtual** Acad::ErrorStatus OnObjectAppended();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: OrderEntitiesByDrawOrder Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Order an array of objects based on their current draw order.

```
Acad::ErrorStatus OrderEntitiesByDrawOrder(  
    AcDbObjectIdArray & ids  
);
```

Parameters	Description
ids	Input/Output object id array.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: PlaceEntitiesInFrontOfDrawOrder Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Place this array of objects at the top of draw order.

```
Acad::ErrorStatus PlaceEntitiesInFrontOfDrawOrder(  
    AcDbObjectIdArray & ids  
);
```

Parameters	Description
ids	Input/Output object id array.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: QueueObjectsForRegenerate Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Puts all queried objects that are part of this element on the regeneration queue.

```
int QueueObjectsForRegenerate() const;
```

Returns

Returns the number of objects queued for regeneration.

Remarks

Your application must trigger regeneration explicitly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: RemoveStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Removes a style from this element.

```
virtual Acad::ErrorStatus RemoveStyle(  
    AcDbObjectId styleRefId  
);
```

Parameters	Description
styleRefId	Input ID of the style reference.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: RemoveStyle Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Removes a style from this element.

```
Acad::ErrorStatus RemoveStyle(  
    const AcDbObjectId& styleRefId,  
    double dScale  
);
```

Parameters	Description
styleRefId	Input ID of the style reference.
dScale	Input scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: SetAcquiredEntities Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the current selection (defined by calling AcquireEntities()).

```
Acad::ErrorStatus SetAcquiredEntities(  
    AcDbObjectIdArray& Ids  
);
```

Parameters	Description
Ids	Input array of object IDs of the acquired objects.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Called by derived Elements during AcquireEntities() to store their acquisition set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)  
AcMapDMElement:: SetAcquisitionCriteria Method  
[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the acquisition criteria for this element.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters	Description
pDataSourceDsc	Input <a href="#">AcMapDMDataSourceDescriptor</a> data-source descriptor.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This function determines which objects become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

[AcMapDMElement:: SetPreviewBlockId Method](#)

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the AcDbObjectId of the block for kPreviewBlock preview entity type. See also GetPreviewBlockId().

```
Acad::ErrorStatus SetPreviewBlockId(  
    AcDbObjectId blockId  
);
```

Parameters	Description
blockId	Input AcDbObjectId of the block.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: SetVisible Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Sets the element visibility at the specified scale. See also IsVisible().

```
virtual Acad::ErrorStatus SetVisible(  
    bool bNewVal,  
    double dScale = 0  
);
```

Parameters	Description
bNewVal	Input true to make the element visible at the specified scale, or false to make it invisible.
dScale	Input scale at which visibility is set. The default value is zero, the current scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: subErase Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

Parameters	Description
erasing	Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMElement Class](#), [AcMapDMElement Class](#)

AcMapDMElement:: UpdateStylization Method

[AcMapDMElement Class](#) | [AcMapDMElement Class](#)

Stylizes the current or updated selection at the current scale.

```
virtual Acad::ErrorStatus UpdateStylization(  
    bool bAcquireEntities  
);
```

Parameters

Description

bAcquireEntities

Input true to acquire entities.

Returns

Returns Acad::eOk if successful. Returns Acad::eAmbiguousOutput if bAcquireEntities is true but the previous selection has not been dismissed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)  
AcMapDMEntityStyle:: ~AcMapDMEntityStyle Destructor  
[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMEntityStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)  
AcMapDMEntityStyle:: AcMapDMEntityStyle Constructor  
[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Constructs an instance of this class.

```
AcMapDMEntityStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: Apply Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Applies the style to an entity.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to apply the style to. This entity must have been opened for write.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: ClearColor Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the color. See also IsColorCleared().

```
Acad::ErrorStatus ClearColor();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: ClearLinetype Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the linetype. See also [IsLinetypeCleared\(\)](#).

```
Acad::ErrorStatus ClearLinetype();
```

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: ClearLinetypeScale Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the linetype scale. See also [IsLinetypeScaleCleared\(\)](#).

```
Acad::ErrorStatus ClearLinetypeScale();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: IsLinetypeScaleCleared Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Determines whether the linetype scale is cleared. See also ClearLinetypeScale().

**bool** IsLinetypeScaleCleared() **const**;

Returns

Returns true if the linetype scale is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: ClearLineWeight Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the lineweight. See also [IsLineWeightCleared\(\)](#).

```
Acad::ErrorStatus ClearLineWeight();
```

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: ClearPlotstyleName Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the plotstyle. See also [IsPlotstyleNameCleared\(\)](#).

```
Acad::ErrorStatus ClearPlotstyleName();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: IsPlotstyleNameCleared Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Determines whether the plotstyle is cleared. See also ClearPlotstyleName().

**bool** IsPlotstyleNameCleared() **const**;

Returns

Returns true if the plotstyle is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: clone Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle::copyFrom Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle::Dismiss Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Clears the stylization on an entity.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: dwgInFields Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: dwgOutFields Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: dxfInFields Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: dxfOutFields Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: GetColor Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the color. See also SetColor().

```
Acad::ErrorStatus GetColor(  
    const AcCmColor*& pColor  
) const;
```

Parameters	Description
pColor	Output color value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: GetLinetype Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the linetype id. See also SetLinetype().

```
Acad::ErrorStatus GetLinetype(  
    AcDbObjectId& linetypeId  
) const;
```

Parameters	Description
linetypeId	Output linetype id.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: GetLinetypeScale Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the linetype scale. See also SetLinetypeScale().

```
Acad::ErrorStatus GetLinetypeScale(  
    double& dLinetypeScale  
) const;
```

Parameters	Description
dLinetypeScale	Output linetype scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: GetLineWeight Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves the lineweight. See also SetLineWeight().

```
Acad::ErrorStatus GetLineWeight(  
    AcDb::Lineweight& lw  
) const;
```

Parameters	Description
lw	Output lineweight.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: IsColorCleared Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Determines whether the color is cleared. See also ClearColor().

```
bool IsColorCleared() const;
```

Returns

Returns true if the color is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: IsLinetypeCleared Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Determines whether the linetype is cleared. See also `ClearLinetype()`.

**bool** IsLinetypeCleared() **const**;

Returns

Returns true if the linetype is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle:: IsLineWeightCleared Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Determines whether the lineweight is cleared. See also ClearLineWeight().

```
bool IsLineWeightCleared() const;
```

Returns

Returns true if the lineweight is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

AcMapDMEntityStyle::SetColor Method

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the color. See also GetColor().

```
Acad::ErrorStatus SetColor(  
    AcCmColor color  
);
```

Parameters	Description
color	Input color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: SetLinetype Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the linetype. See also GetLinetype().

```
Acad::ErrorStatus SetLinetype(  
    const AcDbObjectId linetypeId  
);
```

Parameters	Description
linetypeId	Input linetype id.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: SetLinetypeScale Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the linetype scale. See also GetLinetypeScale().

```
Acad::ErrorStatus SetLinetypeScale(  
    double dLinetypeScale  
);
```

Parameters	Description
dLinetypeScale	Input linetype scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: SetLineWeight Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Sets the linewidth. See also [GetLineWeight\(\)](#).

```
Acad::ErrorStatus SetLineWeight(  
    AcDb::Lineweight lw  
);
```

Parameters	Description
lw	Input linewidth.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: UnApply Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Removes the style from a specified entity.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMEntityStyle Class](#), [AcMapDMEntityStyle Class](#)

[AcMapDMEntityStyle:: Update Method](#)

[AcMapDMEntityStyle Class](#) | [AcMapDMEntityStyle Class](#)

Retrieves and stylizes an entity.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureDataSourceDescriptor Class](#),

[AcMapDMFeatureDataSourceDescriptor Class](#)

AcMapDMFeatureDataSourceDescriptor::

~AcMapDMFeatureDataSourceDescriptor Destructor

[AcMapDMFeatureDataSourceDescriptor Class](#) |

[AcMapDMFeatureDataSourceDescriptor Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMFeatureDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureDataSourceDescriptor Class](#),

[AcMapDMFeatureDataSourceDescriptor Class](#)

AcMapDMFeatureDataSourceDescriptor::

AcMapDMFeatureDataSourceDescriptor Constructor

[AcMapDMFeatureDataSourceDescriptor Class](#) |

[AcMapDMFeatureDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMFeatureDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
[AcMapDMFeatureElement:: ~AcMapDMFeatureElement Destructor](#)  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMFeatureElement ();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
AcMapDMFeatureElement:: AcMapDMFeatureElement Constructor  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Constructs an instance of this class.

```
AcMapDMFeatureElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

[AcMapDMFeatureElement::AcquireEntities Method](#)

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

See also [AcMapDMDataSourceDescriptor::SetDrawingList\(\)](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

AcMapDMFeatureElement:: ClonesObjectsFromExternalSource Method

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

AcMapDMFeatureElement:: dwgInFields Method

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

[AcMapDMFeatureElement:: dwgOutFields Method](#)

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

`pFiler`

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

[AcMapDMFeatureElement:: dxfInFields Method](#)

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

[AcMapDMFeatureElement:: dxfOutFields Method](#)

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

AcMapDMFeatureElement:: GetAcquisitionCriteria Method

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters	Description
pDataSourceDsc	Output pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)  
AcMapDMFeatureElement:: OnMapProjectInitialized Method  
[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMFeatureElement Class](#), [AcMapDMFeatureElement Class](#)

[AcMapDMFeatureElement:: SetAcquisitionCriteria Method](#)

[AcMapDMFeatureElement Class](#) | [AcMapDMFeatureElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters

Description

pDataSourceDsc

Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

[AcMapDMGroup:: ~AcMapDMGroup Destructor](#)

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMGroup();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: AcMapDMGroup Constructor

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Constructs an instance of this class.

```
AcMapDMGroup();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup::audit Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

### Parameters

### Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

### Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup::DismissStylization Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Clears the current stylization.

```
virtual Acad::ErrorStatus DismissStylization(  
    bool bDismissEntities  
);
```

Parameters

Description

bDismissEntities

Input true to clear the acquired entities.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: dwgInFields Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: dwgOutFields Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: dxfInFields Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: dxfOutFields Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: erased Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Lets this object listen to erase-notifications from items that it owns. See also erased() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual void erased(  
    const AcDbObject* dbObj,  
    Adesk::Boolean pErasing = true  
);
```

Parameters	Description
dbObj	Input object that was erased.
pErasing	Input true to listen as an erase is happening. The default value is true.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: HasChildrenInSpecifiedVisibleState Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Determines whether children are visible or invisible at a specified scale.

```
bool HasChildrenInSpecifiedVisibleState(  
    bool bState,  
    double dScale  
);
```

Parameters	Description
bState	Input true to check visibility.
dScale	Input scale.

Returns

Returns true if children are visible at the specified scale; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup::IsEmpty Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Determines whether this group is empty.

```
virtual bool IsEmpty() const;
```

Returns

Returns true if the group is empty.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: isVisible Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Determines whether this group is visible at the specified scale. See also [setVisible\(\)](#) and [HasChildrenInSpecifiedVisibleState\(\)](#).

```
virtual bool isVisible(  
    double dScale  
) const;
```

Parameters

Description

dScale

Input scale at which visibility is evaluated.

Returns

Returns true if the group is visible at the specified scale; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)  
AcMapDMGroup:: NewAllItemsIterator Method  
[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Retrieves a new iterator over the contents of this group.

```
AcMapDMAllItemsIterator* NewAllItemsIterator(  
    bool bForward = true,  
    bool bSkipProxies = false  
);
```

Parameters	Description
bForward	Input true for a forward traversal.
bSkipProxies	Input true to skip proxy Items

## Returns

Returns a new [AcMapDMAllItemsIterator](#) if successful, or NULL if unsuccessful.

## Remarks

This iterator will step on every Item, even if it is in a proxy state, unless true is passed in the bSkipProxies argument.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: OnMapProjectInitialized Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: OnObjectAppended Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Invoked when an object is appended to the database.

**virtual** Acad::ErrorStatus OnObjectAppended();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: RemoveItem Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Removes an item from this group.

```
virtual Acad::ErrorStatus RemoveItem(  
    const AcDbObjectId& Id  
);
```

Parameters

Description

Id

Input ID of the item to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup::SelectElementsWithStyleApplied Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Retrieves the IDs of the elements that have references to the specified style at the specified scale.

```
virtual Acad::ErrorStatus SelectElementsWithStyleApplied(  
    AcDbObjectIdArray& aElementIds,  
    const AcDbObjectId& styleId,  
    double dDisplayScale,  
    bool bSkipDisabled = true  
);
```

Parameters	Description
aElementIds	Output IDs of the elements that refer to the style.
styleId	Input ID of the AcMapDMStyle object.
dDisplayScale	Input display scale.
bSkipDisabled	Input true to skip styles that are disabled at the specified scale. The default value is true.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: SetVisible Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Sets the group visibility at the specified scale. See also [IsVisible\(\)](#) and [HasChildrenInSpecifiedVisibleState\(\)](#).

```
virtual Acad::ErrorStatus SetVisible(  
    bool bNewVal,  
    double dScale  
);
```

Parameters

Description

bNewVal

Input true to make the group visible at the specified scale, or false to make it invisible.

dScale

Input scale at which visibility is set.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: subErase Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMGroup Class](#), [AcMapDMGroup Class](#)

AcMapDMGroup:: UpdateStylization Method

[AcMapDMGroup Class](#) | [AcMapDMGroup Class](#)

Stylizes the current or updated selection at the current scale.

```
virtual Acad::ErrorStatus UpdateStylization(  
    bool bAcquireEntities  
);
```

Parameters

Description

bAcquireEntities

Input true to acquire entities.

Returns

Returns Acad::eOk if successful. Returns Acad::eAmbiguousOutput if bAcquireEntities is true but the previous selection has not been dismissed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#)

AcMapDMItem:: ELegendDetailLevel Enumeration

[AcMapDMItem Class](#)

Enumerates the levels of legend detail.

```
enum ELegendDetailLevel {  
    kLegendExpanded = 1,  
    kLegendDefault = kLegendExpanded,  
    kLegendCompressed
```

```
};
```

File

DmDisplayItem.h

Parameters	Description
kLegendExpanded	Expanded.
kLegendDefault	Default (kLegendExpanded).
kLegendCompressed	Compressed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: ~AcMapDMItem Destructor](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMItem();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: AcMapDMItem Constructor

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Constructs an instance of this class.

```
AcMapDMItem( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: audit Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters

Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)  
AcMapDMItem:: DismissStylization Method  
[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Clears the current stylization.

```
virtual Acad::ErrorStatus DismissStylization(  
    bool bDismissEntities  
);
```

Parameters	Description
bDismissEntities	Input true to clear the acquired entities.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: dwgInFields Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: dwgOutFields Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

`pFiler`

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: dxfInFields Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: dxfOutFields Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: GetName Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Retrieves the name of this item. See also SetName().

```
virtual const ACHAR* GetName() const;
```

Returns

Returns the item name if successful; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: Implementation Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Returns the implementation object.

```
virtual AcMapDMImpItem* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: isVisible Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Determines whether this item is visible at the specified scale. See also [setVisible\(\)](#).

```
virtual bool isVisible(  
    double dScale  
) const;
```

Parameters

Description

dScale

Input scale at which visibility is evaluated.

Returns

Returns true if the element is visible at the specified scale; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)  
AcMapDMItem:: LegendDetailLevel Method  
[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Retrieves the level of legend detail of this item. See also [SetLegendDetailLevel\(\)](#).

```
ELegendDetailLevel LegendDetailLevel() const;
```

Returns

Returns the ELegendDetailLevelvalue.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: SetLegendDetailLevel Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Sets the level of legend detail of this item. See also LegendDetailLevel().

```
void SetLegendDetailLevel(  
    ELegendDetailLevel eLevel  
);
```

Parameters

Description

eLevel

Input ELegendDetailLevelvalue.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: MapId Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Retrieves the map ID.

```
virtual AcDbObjectId MapId() const;
```

Returns

Returns the map ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: MapProject Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Retrieves the AutoCAD Map project.

```
AcMapProject* MapProject();
```

Returns

Returns the AutoCAD Map project.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: OnMapProjectInitialized Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: OnObjectAppended Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Invoked when an object is appended to the database.

**virtual** Acad::ErrorStatus OnObjectAppended();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: SetVisible Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Sets the item visibility at the specified scale. See also [IsVisible\(\)](#).

```
virtual Acad::ErrorStatus SetVisible(  
    bool bNewVal,  
    double dScale  
);
```

Parameters	Description
bNewVal	Input true to make the item visible at the specified scale, or false to make it invisible.
dScale	Input scale at which visibility is set.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem:: subClose Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subClose();
```

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: subErase Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

AcMapDMItem:: UpdateStylization Method

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Stylizes the current or updated selection at the current scale.

```
virtual Acad::ErrorStatus UpdateStylization(  
    bool bAcquireEntities  
);
```

Parameters

Description

bAcquireEntities

Input true to acquire entities.

Returns

Returns Acad::eOk if successful. Returns Acad::eAmbiguousOutput if bAcquireEntities is true but the previous selection has not been dismissed.

Remarks

Do not override this function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMItem Class](#), [AcMapDMItem Class](#)

[AcMapDMItem::wblockClone Method](#)

[AcMapDMItem Class](#) | [AcMapDMItem Class](#)

Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object pOwnerObject. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus wblockClone(  
    AcRxObject* pOwnerObject,  
    AcDbObject*& pClonedObject,  
    AcDbIdMapping& idMap,  
    Adesk::Boolean isPrimary = true  
) const;
```

Parameters	Description
pOwnerObject	Input object to append the clones to.
pClonedObject	Output the cloned object, or NULL if not cloned.
idMap	Input current ID map.
isPrimary	Input true if this object is primary, or false if it is owned.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerDataSourceDescriptor Class](#),

[AcMapDMLayerDataSourceDescriptor Class](#)

AcMapDMLayerDataSourceDescriptor::

~AcMapDMLayerDataSourceDescriptor Destructor

[AcMapDMLayerDataSourceDescriptor Class](#) |

[AcMapDMLayerDataSourceDescriptor Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMLayerDataSourceDescriptor\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerDataSourceDescriptor Class](#),

[AcMapDMLayerDataSourceDescriptor Class](#)

AcMapDMLayerDataSourceDescriptor::

AcMapDMLayerDataSourceDescriptor Constructor

[AcMapDMLayerDataSourceDescriptor Class](#) |

[AcMapDMLayerDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMLayerDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: ~AcMapDMLayerElement Destructor  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMLayerElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: AcMapDMLayerElement Constructor  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Constructs an instance of this class.

```
AcMapDMLayerElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement:: AcquireEntities Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Runs an ADE query against the drawing scope defined in the data-source descriptor.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement::ClonesObjectsFromExternalSource Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement:: dwgInFields Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement:: dwgOutFields Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement:: dxfInFields Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)

AcMapDMLayerElement:: dxfOutFields Method

[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: GetAcquisitionCriteria Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters	Description
pDataSourceDsc	Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: OnMapProjectInitialized Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLayerElement Class](#), [AcMapDMLayerElement Class](#)  
AcMapDMLayerElement:: SetAcquisitionCriteria Method  
[AcMapDMLayerElement Class](#) | [AcMapDMLayerElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters	Description
pDataSourceDsc	Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

[AcMapDMLegend:: ~AcMapDMLegend Destructor](#)

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMLegend();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

AcMapDMLegend:: AcMapDMLegend Constructor

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Constructs an instance of this class.

```
AcMapDMLegend();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

AcMapDMLegend::DismissTableContents Method

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Clears the contents of the legend.

```
virtual Acad::ErrorStatus DismissTableContents();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

AcMapDMLegend:: dwgInFields Method

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

[AcMapDMLegend:: dwgOutFields Method](#)

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
<code>pFiler</code>	Input filer to use to write the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

AcMapDMLegend:: dxfInFields Method

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

[AcMapDMLegend:: dxfOutFields Method](#)

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Lets this object write its data. See also `dxfOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

[AcMapDMLegend:: Implementation Method](#)

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Returns the implementation object.

```
virtual AcMapDMImpLegend* Implementation();
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMLegend Class](#), [AcMapDMLegend Class](#)

[AcMapDMLegend:: UpdateTableContents Method](#)

[AcMapDMLegend Class](#) | [AcMapDMLegend Class](#)

Clears the contents of the legend and updates it with current data from the display manager.

```
virtual Acad::ErrorStatus UpdateTableContents(  
    AcMapDMMap& map,  
    bool bFirstUpdate  
);
```

Parameters	Description
map	Input AcMapDMMap containing the legend.
bFirstUpdate	This parameter indicates how the size of the legend is calculated. A true value will set the width of the legend to be 20% of the current viewport width. A false value will retain the legend's current width. In either case, the height of the legend is determined by the contents loaded from the display manager.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#)

AcMapDMMMap:: DOMode Enumeration

[AcMapDMMMap Class](#)

This is record AcMapDMMMap::DOMode.

```
enum DOMode {  
    kDOTextOnTop = 0x0001,  
    kDOHatchOnBottom = 0x0002  
};  
File
```

DmMap.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: ~AcMapDMMMap Destructor](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMMMap();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: AcMapDMMMap Constructor

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Constructs an instance of this class.

```
AcMapDMMMap( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::audit Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters

Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)  
AcMapDMMap:: CreateLegend Method  
[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Creates a new legend for the specified layout.

```
Acad::ErrorStatus CreateLegend(  
    AcMapDMLegend*& pLegend,  
    AcDbObjectId layoutId = 0  
);
```

Parameters	Description
pLegend	Output created AcMapDMLegend.
layoutId	Input ID of the layout.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

If the legend already exists, it is overwritten.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::deepCloneObject Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Clones this object and all the objects that it refers to.

```
static Acad::ErrorStatus deepCloneObject(  
    AcDbObjectId& newId,  
    AcDbObjectId& existingId  
);
```

Parameters	Description
newId	Output ID of the cloned object.
existingId	Input ID of an existing object.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::deepCloneObjects Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Clones this object and all the objects that it refers to.

```
static Acad::ErrorStatus deepCloneObjects(  
    AcDbIdMapping& idMap,  
    AcDbObjectIdArray& existingIdArray  
);
```

Parameters

Description

idMap

Output ID mapping.

existingIdArray

Input array of IDs of existing objects.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::DeleteLegend Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Deletes the contents of the legend of the specified layout.

```
Acad::ErrorStatus DeleteLegend(  
    AcDbObjectId layoutId = 0  
);
```

Parameters	Description
layoutId	Input ID of the layout.

Returns

Returns Acad::eOk if successful. Returns Acad::eKeyNotFound if no legend exists for the specified layout.

## Remarks

This function deletes the legend itself, and removes the layout from the internal list of layout/legend pairs.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap::DismissStylization Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Clears the current stylization.

```
virtual Acad::ErrorStatus DismissStylization(  
    bool bDismissEntities  
);
```

Parameters	Description
bDismissEntities	Input true to clear the acquired entities.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: dwgInFields Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: dwgOutFields Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: dxfInFields Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: dxfOutFields Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: erased Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Lets this object listen to erase-notifications from items that it owns. See also erased() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual void erased(  
    const AcDbObject* dbObj,  
    Adesk::Boolean pErasing = true  
);
```

Parameters	Description
dbObj	Input object that was erased.
pErasing	Input true to listen as an erase is happening. The default value is true.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap:: GetAllDrawOrderItemsIterator Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Retrieves the draw-order iterator. The first function call iterates elements in the order that they are arranged in the map tree. This iterator may return Items that are in a proxy state, unless the bSkipProxies argument is passed as true. See also IsDrawOrderDefined().

```
AcMapDMAllDrawOrderItemsIterator* GetAllDrawOrderItemsIterator(  
    bool bForward = true,  
    bool bSkipProxies = false  
);
```

Parameters	Description
bForward	Input true to traverse forward. The default value is true.
bSkipProxies	Input true to skip over proxy Items The default value is false;

## Returns

Returns the [AcMapDMAllDrawOrderItemsIterator](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: GetDOMode Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Gets the draw order mode currently in use for the map.

```
int GetDOMode() const;
```

Returns

A bit flag containing the current draw order mode of the map. The bits are one of the values in DOMode enumeration.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap:: GetLegend Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Retrieves the legend of the specified layout.

```
Acad::ErrorStatus GetLegend(  
    AcMapDMLegend*& pLegend,  
    AcDb::OpenMode openMode,  
    AcDbObjectId layoutId = 0  
);
```

Parameters	Description
pLegend	Output AcMapDMLegendobject.
openMode	Input open mode (typically AcDb::kForWrite).
layoutId	Input ID of the layout.

## Returns

Returns Acad::eOk if successful. Returns Acad::eKeyNotFound if no legend exists for the specified layout.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetLegendId Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the legend of the specified layout.

```
Acad::ErrorStatus GetLegendId(  
    AcDbObjectId& legendId,  
    AcDbObjectId layoutId = 0  
);
```

Parameters	Description
legendId	Output ID of the legend.
layoutId	Input ID of the layout.

Returns

Returns Acad::eOk if successful. Returns Acad::eKeyNotFound if no legend exists for the specified layout.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: GetLineTypeScale Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the linetype scale applied to the stylization. The linetype scale, a positive value, is assigned to the AutoCAD system variable LTSCALE at update time. See also SetLineTypeScale().

```
double GetLineTypeScale(  
    double dScale = 0.  
) const;
```

Parameters	Description
dScale	Input threshold scale value.

Returns

Returns the linetype scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetName Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the name of this map. See also SetName().

```
virtual const ACHAR* GetName() const;
```

Returns

Returns the map name if successful; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: GetTableStyle Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the table style. See also [SetTableStyle\(\)](#).

```
const ACHAR* GetTableStyle() const;
```

Returns

Returns the name of the table style.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: GetThumbnailDimensions Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the thumbnail width and height factors. These width and height factors are multiplied by the text height in the legend to determine the size of the thumbnail width and height. See also [SetThumbnailDimensions\(\)](#).

```
void GetThumbnailDimensions(  
    double& dHeight,  
    double& dWidth  
) const;
```

Parameters	Description
dHeight	Output height factor.
dWidth	Output width factor.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap:: SetThumbnailDimensions Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Sets the thumbnail width and height factors. These width and height factors are multiplied by the text height in the legend to determine the size of the thumbnail width and height. See also [GetThumbnailDimensions\(\)](#).

```
Acad::ErrorStatus SetThumbnailDimensions(  
    double dHeight,  
    double dwidth  
);
```

Parameters	Description
dHeight	Input height factor.
dWidth	Input width factor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: GetThumbnailFraming Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Determines whether a box is drawn around the thumbnail graphic. See also [SetThumbnailFraming\(\)](#).

```
bool GetThumbnailFraming() const;
```

Returns

Returns true if a box is drawn.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: IsDrawOrderDefined Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Determines whether the draw order is defined. Draw order is defined on demand when `GetDrawOrderIterator()` is first called. Draw-order functionality is available in Display Manager user interface in the list view.

```
bool IsDrawOrderDefined() const;
```

Returns

Returns true if draw order is defined.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: IsLinkToFileEnabled Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Determines whether linked-file capability is enabled.

```
bool IsLinkToFileEnabled(  
    double dScale = 0.  
) const;
```

Parameters	Description
dScale	Input threshold scale value.

Returns

Returns true if the capability is enabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: IsStylized Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Determines whether this map is stylized.

```
bool IsStylized() const;
```

Returns

Returns true if the map is stylized.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

[AcMapDMMap:: isVisible Method](#)

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Determines whether this map is visible at the specified scale. See also [setVisible\(\)](#).

```
virtual bool isVisible(  
    double dScale  
) const;
```

Parameters

Description

dScale

Input scale threshold at which visibility is evaluated.

Returns

Returns true if the map is visible at the specified scale; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: MapId Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Retrieves the ID of this map.

```
virtual AcDbObjectId MapId() const;
```

Returns

Returns the map ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: OnMapProjectInitialized Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: OnObjectAppended Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Invoked when an object is appended to the database.

**virtual** Acad::ErrorStatus OnObjectAppended();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)  
AcMapDMMap:: OnStyleModified Method  
[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Makes the system to refresh the contents of map when the given style is modified.

```
Acad::ErrorStatus OnStyleModified(  
    AcDbObjectId styleId  
);
```

Parameters	Description
styleId	Input id of modified style.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This method is meant to be called by an application whenever a custom style is modified so that the system could invoke appropriate methods on the custom style to allow it to update itself.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: Requery Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Requeries objects belonging to the map.

```
Acad::ErrorStatus Requery();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This method erases previous selection and requeries to obtain new set of objects.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap::ResumeFixingDO Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Resumes fixing of the draw order of entities. The draw order is fixed during this call. Any change that affects the draw order after this method is called until the next SuspendFixingDO method will be reflected immediately. This method **must** be used in conjunction with SuspendFixingDO() and these two methods must always be paired.

```
int ResumeFixingDO(  
    bool bFixDO = true  
);
```

Parameters

Description

bFixDO

Input true if draw order must be fixed.

Returns

Returns the outstanding reference counts before fixing draw order can be resumed. If bFixDO is true, the draw order will be fixed right away if SuspendFixingDO/ResumeFixingDO calls match. If it's false, then the reference count maintained to track Suspend/Resume calls is decremented by one and draw order is not fixed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SaveLinkedFile Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Saves the current database in its current state of stylization at the current scale to the filename specified in a previous call to [AcMapDMMMap::SetLinkedFileName\(\)](#). The linked file contains a stylized map that can be plotted or published to DWF. Clicking the Update button in the AutoCAD Map user interface invokes SaveLinkedFile(). SaveLinkedFile() does not call [AcMapDMMMap::UpdateStylization\(\)](#); it is your application's responsibility to prepare the stylized model. SaveLinkedFile() calls AcMapDMMMap::GetCurrentScale(), followed by [AcMapDMMMap::GetLinkedFileName\(\)](#), with the current scale to obtain the target filename.

```
Acad::ErrorStatus SaveLinkedFile(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: UpdateStylization Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Stylizes the current or updated selection at the current scale.

```
virtual Acad::ErrorStatus UpdateStylization(  
    bool bAcquireEntities  
);
```

Parameters

Description

bAcquireEntities

Input true to acquire entities.

Returns

Returns Acad::eOk if successful. Returns Acad::eAmbiguousOutput if bAcquireEntities is true but the previous selection has not been dismissed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

AcMapDMMap::SelectElementsWithStyleApplied Method

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Retrieves elements that reference the specified style at the specified scale threshold.

```
virtual Acad::ErrorStatus SelectElementswithStyleApplied(  
    AcDbObjectIdArray& aElementIds,  
    const AcDbObjectId& styleId,  
    double dDisplayScale,  
    bool bSkipDisabled = true  
);
```

Parameters	Description
aElementIds	Output IDs of the elements that refer to the style.
styleId	Input AcMapDMStyle object ID.
dDisplayScale	Input display scale.
bSkipDisabled	Input true to skip styles that are disabled at the specified scale. The default value is true.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: SetDOMode Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the draw order mode for the map.

```
void SetDOMode(  
    int mode  
);
```

Parameters

Description

mode

A bit flag containing the desired draw order mode.

Returns

Nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: SetLineStyleScale Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the linetype scale for a specified scale. See also GetLineStyleScale().

```
Acad::ErrorStatus SetLineStyleScale(  
    double sLineStyleScale,  
    double dScale = 0.  
);
```

Parameters	Description
sLineStyleScale	Input linetype scale to set.
dScale	Input threshold scale value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: SetLinkToFileEnabled Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets the status of the linked file capability.

```
Acad::ErrorStatus SetLinkToFileEnabled(  
    bool bNewValue,  
    double dScale = 0.  
);
```

Parameters	Description
bNewValue	Input true to enable, or false to disable, the capability.
dScale	Input threshold scale value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

[AcMapDMMMap:: SetThumbnailFraming Method](#)

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Sets whether a box is drawn around the thumbnail graphic. See also [GetThumbnailFraming\(\)](#).

```
void SetThumbnailFraming(  
    bool bDrawFrame  
);
```

Parameters	Description
bDrawFrame	Input true to draw a box.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMap Class](#), [AcMapDMMap Class](#)

[AcMapDMMap:: SetVisible Method](#)

[AcMapDMMap Class](#) | [AcMapDMMap Class](#)

Sets the map visibility at the specified scale. See also [IsVisible\(\)](#).

```
virtual Acad::ErrorStatus SetVisible(  
    bool NewVal,  
    double dScale  
);
```

Parameters	Description
NewVal	Input true to make the map visible at the specified scale, or false to make it invisible.
dScale	Input scale threshold at which visibility is set.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)

AcMapDMMMap:: subErase Method

[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMap Class](#), [AcMapDMMMap Class](#)  
AcMapDMMMap:: SuspendFixingDO Method  
[AcMapDMMMap Class](#) | [AcMapDMMMap Class](#)

Suspends fixing of the draw order of entities.

```
int SuspendFixingDO();
```

Returns

Returns the outstanding reference counts before fixing DO can be resumed.

Remarks

Any change in draw order after this call is made is not effected until ResumeFixingDO method is called. Users of this method are strongly advised to correctly pair the Suspend/Resume calls. If this method is not called (later followed by ResumeFixingDO), then any change in DO is immediately fixed and displayed. This method is recommended for use during changes (addition, moving, / etc.) involving multiple items.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapIterator Class](#), [AcMapDMMMapIterator Class](#)

[AcMapDMMMapIterator:: ~AcMapDMMMapIterator Destructor](#)

[AcMapDMMMapIterator Class](#) | [AcMapDMMMapIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMMMapIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapIterator Class](#), [AcMapDMMapIterator Class](#)

AcMapDMMapIterator:: Done Method

[AcMapDMMapIterator Class](#) | [AcMapDMMapIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() const = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapIterator Class](#), [AcMapDMMMapIterator Class](#)

AcMapDMMMapIterator:: GetObj Method

[AcMapDMMMapIterator Class](#) | [AcMapDMMMapIterator Class](#)

Retrieves the current object in the iteration.

```
virtual Acad::ErrorStatus GetObj(  
    AcMapDMMMap*& pMap,  
    AcDb::OpenMode mode  
) const = 0;
```

Parameters	Description
pMap	Output pointer to the current AcMapDMMMap.
mode	Input mode for opening the object.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapIterator Class](#), [AcMapDMMMapIterator Class](#)

[AcMapDMMMapIterator:: Next Method](#)

[AcMapDMMMapIterator Class](#) | [AcMapDMMMapIterator Class](#)

Advances to the next element in the iteration.

```
virtual bool Next() = 0;
```

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapIterator Class](#), [AcMapDMMMapIterator Class](#)

[AcMapDMMMapIterator:: ObjectId Method](#)

[AcMapDMMMapIterator Class](#) | [AcMapDMMMapIterator Class](#)

Retrieves the ID of the current object in the iteration.

```
virtual AcDbObjectId ObjectId() const = 0;
```

Returns

Returns a valid ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)  
AcMapDMMMapManager:: ~AcMapDMMMapManager Destructor  
[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMMMapManager();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)  
AcMapDMMMapManager:: AcMapDMMMapManager Constructor  
[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Constructs an instance of this class.

```
AcMapDMMMapManager ( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: AddDefaultMap Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Creates a default map object.

```
Acad::ErrorStatus AddDefaultMap(  
    AcDbObjectId& Id  
);
```

Parameters

Description

Id

Output ID of the default map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: CreateNewMap Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Adds a new map with the default settings.

```
Acad::ErrorStatus CreateNewMap(  
    AcDbObjectId& Id  
);
```

Parameters	Description
Id	Output ID of the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

The new map is named "New\_Map#", where # is a serial number starting at 1.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: dwgInFields Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: dwgOutFields Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: dxfInFields Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: dxfOutFields Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager::GetCurrent Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Retrieves the ID of the current map.

```
Acad::ErrorStatus GetCurrent(  
    AcDbObjectId& Id  
) const;
```

Parameters

Description

Id

Output ID of the current map ID.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Has Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Determines whether a map with the specified ID exists.

```
bool Has(  
    AcDbObjectId objId  
) const;
```

Parameters	Description
objId	Input ID of the map to search for.

Returns

Returns true if the map exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: Implementation Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Returns the implementation object.

```
AcMapDMImpMapManager* Implementation();
```

Returns

Returns the implementation object, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: NewIterator Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Returns a new map iterator.

```
AcMapDMMMapIterator* NewIterator();
```

Returns

Returns an AcMapDMMMapIterator if successful; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

[AcMapDMMMapManager:: NumMaps Method](#)

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Counts the number of maps.

```
Adesk::UInt32 NumMaps() const;
```

Returns

Returns the number of maps.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMapManager Class](#), [AcMapDMMapManager Class](#)

AcMapDMMapManager:: Remove Method

[AcMapDMMapManager Class](#) | [AcMapDMMapManager Class](#)

Removes the map with the specified ID.

```
Acad::ErrorStatus Remove(  
    AcDbObjectId Id  
);
```

Parameters	Description
Id	Input ID of the map to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The default map cannot be removed. If the current map is removed, the default map becomes the current map.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapManager Class](#), [AcMapDMMMapManager Class](#)

AcMapDMMMapManager:: SetCurrent Method

[AcMapDMMMapManager Class](#) | [AcMapDMMMapManager Class](#)

Sets the map with the specified ID to be the current map.

```
Acad::ErrorStatus SetCurrent(  
    const AcDbObjectId& Id  
);
```

Parameters	Description
Id	Input ID of the map.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapManager Class](#), [AcMapDMMapManager Class](#)

AcMapDMMapManager:: wblockClone Method

[AcMapDMMapManager Class](#) | [AcMapDMMapManager Class](#)

Clones the specified map to the destination database. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
static Acad::ErrorStatus wblockClone(  
    const AcDbObjectId& mapId,  
    AcDbDatabase* pDestDb  
);
```

Parameters	Description
mapId	Input map ID.
pDestDb	Input destination database.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMMapManager Class](#), [AcMapDMMapManager Class](#)

[AcMapDMMapManager::wblockCloneObjects Method](#)

[AcMapDMMapManager Class](#) | [AcMapDMMapManager Class](#)

Deep clones the specified objects and appends them to the specified container. The objects can come from multiple source databases, and must match the type of owner specified, but must be from a different database than the ownerId object. See also `wblockCloneObjects()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
static Acad::ErrorStatus wblockCloneObjects(  
    AcDbObjectIdArray& objectIds,  
    AcDbObjectId& ownerId,  
    AcDbIdMapping& idMap,  
    AcDb::DuplicateRecordCloning drc,  
    bool deferXlation = false  
);
```

Parameters	Description
objectIds	Input array of IDs of objects to clone. This array can contain only AcMapDMItem-derived objects (AcMapDmElement and AcMapDMGroup instances, for example).
ownerId	Input ID of the container object that will own the clones. This ID can refer to only an AcMapDMGroup-derived object in the destination database.
idMap	Input array of AcDbIdPair objects to use to translate object ID relationships.
drc	Input action for duplicate records.
deferXlation	Input true to defer ID translation. The default value is false.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: CurrentScaleModified Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when the current scale is modified.

```
virtual void CurrentScaleModified(  
    const AcMapDMMMap* pMap,  
    double dNewScale,  
    double dOldScale  
);
```

Parameters	Description
pMap	Input pointer to the AcMapDMMMapobject.
dNewScale	Input value of new scale.
dOldScale	Input value of the old scale.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::DismissStylizationBegin Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked just before stylization is dismissed.

```
virtual void DismissStylizationBegin(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::DismissStylizationCancel Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a stylization dismissal is cancelled.

```
virtual void DismissStylizationCancel(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::DismissStylizationEnd Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked just after stylization is dismissed.

```
virtual void DismissStylizationEnd(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::ItemAppended Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a display-management item is added to the map.

```
virtual void ItemAppended(  
    const AcMapDMItem * pItem  
);
```

Parameters

Description

pItem

Input pointer to the added AcMapDMItemobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::ItemErased Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a display-management item is erased from the map.

```
virtual void ItemErased(  
    const AcMapDMItem * pItem,  
    bool bErased  
);
```

Parameters	Description
pItem	Input pointer to the erased AcMapDMItemobject.
bErased	Input true to erase the item.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::ItemModified Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a display-management item is modified.

```
virtual void ItemModified(  
    const AcMapDMItem * pItem  
);
```

Parameters

Description

pItem

Input pointer to the modified AcMapDMItemobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: ScaleAdded Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a scale is added.

```
virtual void ScaleAdded(  
    double dScale,  
    bool bIsCopy  
);
```

Parameters

Description

dScale

Input scale value.

bIsCopy

Input true if the new scale is added by copying styles.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: ScaleErased Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a scale is erased.

```
virtual void ScaleErased(  
    double dScale  
);
```

Parameters	Description
dScale	Input scale value.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: ScaleModified Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a scale is modified.

```
virtual void ScaleModified(  
    double dNewScale,  
    double dOldScale  
);
```

Parameters	Description
dNewScale	Input value of the new scale.
dOldScale	Input value of the old scale.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor:: StyleAppended Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style is added.

```
virtual void StyleAppended(  
    const AcMapDMStyle* pStyle  
);
```

Parameters

Description

pStyle

Input pointer to the added AcMapDMStyleobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor:: StyleErased Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style is erased.

```
virtual void StyleErased(  
    const AcMapDMStyle* pStyle,  
    bool bErased  
);
```

Parameters	Description
pStyle	Input pointer to the erased AcMapDMStyleobject.
bErased	Input true to erase the style.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor:: StyleModified Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style is modified.

```
virtual void StyleModified(  
    const AcMapDMStyle* pStyle  
);
```

Parameters

Description

pStyle

Input pointer to the modified AcMapDMStyleobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor::StyleReferenceAppended Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style reference is added.

```
virtual void StyleReferenceAppended(  
    const AcMapDMElement* pElement,  
    const AcMapDMStyleReference* pStyleRef,  
    double dScale  
);
```

Parameters	Description
pElement	Input pointer to the AcMapDMElementobject that the style is added to.
pStyleRef	Input pointer to the added <a href="#">AcMapDMStyleReference</a> object.
dScale	Input scale at which the style reference is applied.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor:: StyleReferenceErased Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style reference is erased.

```
virtual void StyleReferenceErased(  
    const AcMapDMStyleReference* pStyleRef,  
    bool bErased  
);
```

Parameters

Description

pStyleRef

Input pointer to the erased  
[AcMapDMStyleReference](#) object.

bErased

Input true to erase the style reference.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMapReactor Class](#), [AcMapDMMapReactor Class](#)

AcMapDMMapReactor:: StyleReferenceModified Method

[AcMapDMMapReactor Class](#) | [AcMapDMMapReactor Class](#)

Invoked when a style reference is modified.

```
virtual void StyleReferenceModified(  
    const AcMapDMStyleReference* pStyleRef  
);
```

Parameters

Description

pStyleRef

Input pointer to the modified  
[AcMapDMStyleReference](#) object.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: UpdateStylizationBegin Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked just before stylization occurs.

```
virtual void UpdateStylizationBegin(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor:: UpdateStylizationCancel Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked when a stylization is cancelled.

```
virtual void UpdateStylizationCancel(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMMMapReactor Class](#), [AcMapDMMMapReactor Class](#)

AcMapDMMMapReactor::UpdateStylizationEnd Method

[AcMapDMMMapReactor Class](#) | [AcMapDMMMapReactor Class](#)

Invoked just after stylization occurs.

```
virtual void UpdateStylizationEnd(  
    const AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: CategoryAppended Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a category is added to the style library.

```
virtual void CategoryAppended(  
    const AcMapDMStyleLibrary* pStyleLibrary,  
    const AcMapDMStyleCategory * pCategory  
);
```

Parameters	Description
pStyleLibrary	Input pointer to the AcMapDMStyleLibraryobject.
pCategory	Input pointer to the <a href="#">AcMapDMStyleCategory</a> object.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: CategoryModified Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a category is modified.

```
virtual void CategoryModified(  
    const AcMapDMStyleCategory * pCategory  
);
```

Parameters	Description
pCategory	Input pointer to the <a href="#">AcMapDMStyleCategory</a> object.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: CategoryUnAppended Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a category is detached from the style library.

```
virtual void CategoryUnAppended(  
    const AcMapDMStyleLibrary* pStyleLibrary,  
    const AcMapDMStyleCategory * pCategory,  
    bool bErased  
);
```

Parameters	Description
pStyleLibrary	Input pointer to the AcMapDMStyleLibraryobject.
pCategory	Input pointer to the <a href="#">AcMapDMStyleCategory</a> object.
bErased	Input true to erase the category as well.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: MapAppended Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a new map is appended or after a reactor is attached to a project that has map(s) defined.

```
virtual void MapAppended(  
    AcMapDMMap* pMap  
);
```

Parameters	Description
pMap	Input pointer to a AcMapDMMapobject.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)

AcMapDMProjectReactor:: MapGoodBye Method

[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a map is unloaded.

```
virtual void MapGoodBye(  
    AcMapDMMMap* pMap  
);
```

Parameters

Description

pMap

Input pointer to the unloaded AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: MapSetCurrentBegin Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked just before a map is set as the current map.

```
virtual void MapSetCurrentBegin(  
    AcMapDMMMap* pOldCurrentMap  
);
```

Parameters	Description
pOldCurrentMap	Input pointer to the previous AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: MapSetCurrentEnd Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked just after a map is set as the current map.

```
virtual void MapSetCurrentEnd(  
    AcMapDMMMap* pNewCurrentMap  
);
```

Parameters

Description

pNewCurrentMap

Input pointer to the new current AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)

[AcMapDMProjectReactor:: MapSetCurrentFails Method](#)

[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when setting a map as current fails.

```
virtual void MapSetCurrentFails(  
    AcMapDMMMap* pOldCurrentMap  
);
```

Parameters

Description

pOldCurrentMap

Input pointer to the previous AcMapDMMMapobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: MapUnAppended Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a map is detached.

```
virtual void MapUnAppended(  
    AcMapDMMMap* pMap,  
    bool bErased  
);
```

Parameters	Description
pMap	Input pointer to the AcMapDMMMapobject.
bErased	Input true to erase the map as well.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)  
AcMapDMProjectReactor:: StyleAppended Method  
[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a style is added to the category.

```
virtual void StyleAppended(  
    const AcMapDMStyleCategory * pCategory,  
    const AcMapDMStyle* pStyle  
);
```

Parameters	Description
pCategory	Input pointer to the <a href="#">AcMapDMStyleCategory</a> object.
pStyle	Input pointer to the AcMapDMStyle object.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)

[AcMapDMProjectReactor:: StyleModified Method](#)

[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a style is modified.

```
virtual void StyleModified(  
    const AcMapDMStyle* pStyle  
);
```

Parameters	Description
pStyle	Input pointer to the AcMapDMStyleobject.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMProjectReactor Class](#), [AcMapDMProjectReactor Class](#)

AcMapDMProjectReactor:: StyleUnAppended Method

[AcMapDMProjectReactor Class](#) | [AcMapDMProjectReactor Class](#)

Invoked when a style is detached.

```
virtual void StyleUnAppended(  
    const AcMapDMStyle* pStyle,  
    bool bErased  
);
```

Parameters	Description
pStyle	Input pointer to the AcMapDMStyleobject.
bErased	Input true to erase the style as well.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor::

~AcMapDMQueryDataSourceDescriptor Destructor

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMQueryDataSourceDescriptor\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor::

AcMapDMQueryDataSourceDescriptor Constructor

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMQueryDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor:: GetQuery Method

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Retrieves the query definition.

```
virtual Acad::ErrorStatus GetQuery(  
    struct resbuf*& pRb  
) const;
```

Parameters

Description

pRb

Output resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMQueryDataSourceDescriptor Class](#),

[AcMapDMQueryDataSourceDescriptor Class](#)

AcMapDMQueryDataSourceDescriptor:: SetQuery Method

[AcMapDMQueryDataSourceDescriptor Class](#) |

[AcMapDMQueryDataSourceDescriptor Class](#)

Sets the query definition.

```
virtual Acad::ErrorStatus SetQuery(  
    const struct resbuf* pRb  
);
```

Parameters

Description

pRb

Input resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterDataSourceDescriptor Class](#),

[AcMapDMRasterDataSourceDescriptor Class](#)

AcMapDMRasterDataSourceDescriptor::

~AcMapDMRasterDataSourceDescriptor Destructor

[AcMapDMRasterDataSourceDescriptor Class](#) |

[AcMapDMRasterDataSourceDescriptor Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMRasterDataSourceDescriptor\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterDataSourceDescriptor Class](#),

[AcMapDMRasterDataSourceDescriptor Class](#)

AcMapDMRasterDataSourceDescriptor::

AcMapDMRasterDataSourceDescriptor Constructor

[AcMapDMRasterDataSourceDescriptor Class](#) |

[AcMapDMRasterDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMRasterDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: ~AcMapDMRasterElement Destructor  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMRasterElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: AcMapDMRasterElement Constructor  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Constructs an instance of this class.

```
AcMapDMRasterElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

AcMapDMRasterElement:: AcquireEntities Method

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Runs the query against the current drawing.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

AcMapDMRasterElement:: ClonesObjectsFromExternalSource Method

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Returns false (because objects from only the current drawing are selected).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

AcMapDMRasterElement:: dwgInFields Method

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

AcMapDMRasterElement:: dwgOutFields Method

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

AcMapDMRasterElement:: dxfInFields Method

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)

[AcMapDMRasterElement:: dxfOutFields Method](#)

[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: GetAcquisitionCriteria Method  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters	Description
pDataSourceDsc	Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterElement Class](#), [AcMapDMRasterElement Class](#)  
AcMapDMRasterElement:: SetAcquisitionCriteria Method  
[AcMapDMRasterElement Class](#) | [AcMapDMRasterElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters	Description
pDataSourceDsc	Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)  
AcMapDMRasterStyle::~~AcMapDMRasterStyle Destructor  
[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMRasterStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: AcMapDMRasterStyle Constructor](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Constructs an instance of this class.

```
AcMapDMRasterStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle:: Apply Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Applies a style to a raster.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to apply the style to.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: ClearBrightness Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clears image brightness. See also [IsBrightnessCleared\(\)](#).

```
Acad::ErrorStatus ClearBrightness();
```

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: ClearContrast Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clears image contrast. See also [IsContrastCleared\(\)](#).

```
Acad::ErrorStatus ClearContrast();
```

## Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: ClearFade Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clears image fade. See also [IsFadeCleared\(\)](#).

```
Acad::ErrorStatus ClearFade();
```

## Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: ClearTransparency Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clears image transparency. See also [IsTransparencyCleared\(\)](#).

```
Acad::ErrorStatus ClearTransparency();
```

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)  
AcMapDMRasterStyle:: IsTransparencyCleared Method  
[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Determines whether image transparency is cleared. See also `ClearTransparency()`.

**bool** IsTransparencyCleared() **const**;

Returns

Returns true if image transparency is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: clone Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle::copyFrom Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: DeleteCookie Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input cookie to delete.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle::Dismiss Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Clears the stylization of a raster.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle:: dwgInFields Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: dwgOutFields Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: dxfInFields Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle:: dxfOutFields Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle:: Enable Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Enables or disables the style of a raster.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: GetBrightness Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Retrieves image brightness. See also [SetBrightness\(\)](#).

```
Acad::ErrorStatus GetBrightness(  
    Adesk::Int8& brightness  
) const;
```

Parameters	Description
brightness	Output brightness value.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: GetContrast Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Retrieves image contrast. See also SetContrast().

```
Acad::ErrorStatus GetContrast(  
    Adesk::Int8& contrast  
) const;
```

Parameters	Description
contrast	Output contrast value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: GetFade Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Retrieves image fade. See also SetFade().

```
Acad::ErrorStatus GetFade(  
    Adesk::Int8& fade  
) const;
```

Parameters	Description
fade	Output fade value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)  
AcMapDMRasterStyle:: GetTransparency Method  
[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Retrieves image transparency. See also SetTransparency().

```
Acad::ErrorStatus GetTransparency(  
    bool& bTransp  
) const;
```

Parameters	Description
bTransp	Output transparency value.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

AcMapDMRasterStyle:: IsBrightnessCleared Method

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Determines whether image brightness is cleared. See also ClearBrightness().

**bool** IsBrightnessCleared() **const**;

Returns

Returns true if image brightness is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: IsContrastCleared Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Determines whether image contrast is cleared. See also [ClearContrast\(\)](#).

**bool** IsContrastCleared() **const**;

Returns

Returns true if image contrast is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: IsFadeCleared Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Determines whether image fade is cleared. See also [ClearFade\(\)](#).

**bool** IsFadeCleared() **const**;

Returns

Returns true if image fade is cleared.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: SetBrightness Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Sets image brightness. See also `GetBrightness()`.

```
Acad::ErrorStatus SetBrightness(  
    Adesk::Int8 brightness  
);
```

Parameters	Description
brightness	Input brightness value.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: SetContrast Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Sets image contrast. See also `GetContrast()`.

```
Acad::ErrorStatus SetContrast(  
    Adesk::Int8 contrast  
);
```

Parameters	Description
contrast	Input contrast value.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: SetFade Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Sets image fade. See also GetFade().

```
Acad::ErrorStatus SetFade(  
    Adesk::Int8 fade  
);
```

Parameters	Description
fade	Input fade value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: SetTransparency Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Sets image transparency. See also `GetTransparency()`.

```
Acad::ErrorStatus SetTransparency(  
    bool bTransp  
);
```

Parameters	Description
bTransp	Input transparency value.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: UnApply Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Removes the style from a raster.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMRasterStyle Class](#), [AcMapDMRasterStyle Class](#)

[AcMapDMRasterStyle:: Update Method](#)

[AcMapDMRasterStyle Class](#) | [AcMapDMRasterStyle Class](#)

Retrieves and stylizes a raster.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: ~AcMapDMSEAnnotationStyle Destructor](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMSEAnnotationStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: AcMapDMSEAnnotationStyle Constructor](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Constructs an instance of this class.

```
AcMapDMSEAnnotationStyle(  
    Adesk::Boolean bCreateAsSymbolStyle = Adesk::kFalse  
);
```

Parameters	Description
bCreateAsSymbolStyle	If kTrue, then this annotation style is marked as a symbol style.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: clone Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: ColorIsExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the color is an expression.

```
bool ColorIsExpression() const;
```

Returns

Returns true if the color is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle::copyFrom Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: DeleteCookie Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input opaque memory value.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle::Dismiss Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Removes stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
id	Input object ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: dwgInFields Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: dwgOutFields Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

`pFiler`

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: dxfInFields Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: dxfOutFields Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: Enable Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Enables or disables the style for the specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
id	Input object ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: GetAnnotationTemplate Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the annotation template ID.

```
Acad::ErrorStatus GetAnnotationTemplate(  
    AcDbObjectId& templateID  
) const;
```

Parameters	Description
templateID	Output annotation template ID.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetColor Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the color.

```
Acad::ErrorStatus GetColor(  
    AcCmColor& color  
) const;
```

Parameters	Description
color	Output color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: GetLayer Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the layer id.

```
Acad::ErrorStatus GetLayer(  
    AcDbObjectId& layerId  
) const;
```

Parameters	Description
layerId	Output layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetLinetype Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the linetype id.

```
Acad::ErrorStatus GetLinetype(  
    AcDbObjectId& linetypeId  
) const;
```

Parameters	Description
linetypeId	Output linetype AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetLineWeight Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the lineweight.

```
Acad::ErrorStatus GetLineWeight(  
    AcDb::Lineweight& lw  
) const;
```

Parameters	Description
lw	Output lineweight.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetRotation Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the rotation.

```
Acad::ErrorStatus GetRotation(  
    double& dRotation  
) const;
```

Parameters	Description
dRotation	Output rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetScale Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the scale.

```
Acad::ErrorStatus GetScale(  
    double& dScale  
) const;
```

Parameters	Description
dScale	Output scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: GetStylizationEntities Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Retrieves the AcDbObjectIds of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input opaque memory value.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: IsSymbolStyle Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Indicates that this style is a symbol style instead of an annotation style.

```
Adesk::Boolean IsSymbolStyle() const;
```

Returns

Returns kTrue if this is a symbol style; otherwise, returns kFalse.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: LayerIsExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the layer is an expression.

```
bool LayerIsExpression() const;
```

Returns

Returns true if the layer is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: LinetypeIsExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the linetype is an expression.

**bool** LinetypeIsExpression() **const**;

Returns

Returns true if the linetype is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: LineWeightIsExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the linewidth is an expression.

```
bool LineWeightIsExpression() const;
```

Returns

Returns true if the linewidth is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: RotationIsExpression Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the rotation is an expression.

```
bool RotationIsExpression() const;
```

Returns

Returns true if the rotation is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: ScaleIsExpression Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Determines whether the scale is an expression.

```
bool ScaleIsExpression() const;
```

Returns

Returns true if the scale is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)  
[AcMapDMSEAnnotationStyle:: SetAnnotationTemplate Method](#)  
[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the annotation template ID or block ID.

```
Acad::ErrorStatus SetAnnotationTemplate(  
    AcdBObjectId annotationTemplateID  
);
```

Parameters	Description
annotationTemplateID	Input annotation template ID or block ID.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle::SetColor Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the color.

```
Acad::ErrorStatus SetColor(  
    const AcCmColor& color  
);
```

Parameters	Description
color	Input color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

AcMapDMSEAnnotationStyle:: SetLayer Method

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the layer id.

```
Acad::ErrorStatus SetLayer(  
    const AcDbObjectId layerId  
);
```

Parameters	Description
layerId	Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetLinetype Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the linetype id.

```
Acad::ErrorStatus SetLinetype(  
    const AcDbObjectId linetypeId  
);
```

Parameters	Description
linetypeId	Input linetype AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetLineWeight Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the lineweight.

```
Acad::ErrorStatus SetLineWeight(  
    AcDb::Lineweight lw  
);
```

Parameters	Description
lw	Input lineweight.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetRotation Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the rotation.

```
Acad::ErrorStatus SetRotation(  
    double dRotation  
);
```

Parameters	Description
dRotation	Input rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: SetScale Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Sets the scale.

```
Acad::ErrorStatus SetScale(  
    double dScale  
);
```

Parameters	Description
dScale	Input scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEAnnotationStyle Class](#), [AcMapDMSEAnnotationStyle Class](#)

[AcMapDMSEAnnotationStyle:: Update Method](#)

[AcMapDMSEAnnotationStyle Class](#) | [AcMapDMSEAnnotationStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
id	Input object ID.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: ~AcMapDMSEHatchStyle Destructor](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMSEHatchStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)  
AcMapDMSEHatchStyle:: AcMapDMSEHatchStyle Constructor  
[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Constructs an instance of this class.

```
AcMapDMSEHatchStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: clone Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: ColorIsExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Determines whether the color is an expression.

**bool** ColorIsExpression() **const**;

Returns

Returns true if the color is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle::copyFrom Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: DeleteCookie Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input opaque memory value.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle::Dismiss Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Removes stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
id	Input object ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: dwgInFields Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: dwgOutFields Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: dxfInFields Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: dxfOutFields Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: Enable Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Enables or disables the style for the specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
id	Input object ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetColor Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the color.

```
Acad::ErrorStatus GetColor(  
    AcCmColor& color  
) const;
```

Parameters	Description
color	Output color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetLayer Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the layer id.

```
Acad::ErrorStatus GetLayer(  
    AcDbObjectId& layerId  
) const;
```

Parameters	Description
layerId	Output layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: GetRotation Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the rotation.

```
Acad::ErrorStatus GetRotation(  
    double& dRotation  
) const;
```

Parameters	Description
dRotation	Output rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetScale Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the scale.

```
Acad::ErrorStatus GetScale(  
    double& dScale  
) const;
```

Parameters	Description
dScale	Output scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: GetStylizationEntities Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Retrieves the AcDbObjectIds of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input opaque memory value.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: LayerIsExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Determines whether the layer is an expression.

```
bool LayerIsExpression() const;
```

Returns

Returns true if the layer is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle::RotationIsExpression Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Determines whether the rotation is an expression.

**bool** RotationIsExpression() **const**;

Returns

Returns true if the rotation is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: ScaleIsExpression Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Determines whether the scale is an expression.

```
bool ScaleIsExpression() const;
```

Returns

Returns true if the scale is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetColor Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the color.

```
Acad::ErrorStatus SetColor(  
    const AcCmColor& color  
);
```

Parameters	Description
color	Input color value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetLayer Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the layer id.

```
Acad::ErrorStatus SetLayer(  
    const AcDbObjectId layerId  
);
```

Parameters	Description
layerId	Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

AcMapDMSEHatchStyle:: SetRotation Method

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the rotation.

```
Acad::ErrorStatus SetRotation(  
    double dRotation  
);
```

Parameters	Description
dRotation	Input rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: SetScale Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Sets the scale.

```
Acad::ErrorStatus SetScale(  
    double dScale  
);
```

Parameters	Description
dScale	Input scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSEHatchStyle Class](#), [AcMapDMSEHatchStyle Class](#)

[AcMapDMSEHatchStyle:: Update Method](#)

[AcMapDMSEHatchStyle Class](#) | [AcMapDMSEHatchStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
id	Input object ID.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: Justification Enumeration

[AcMapDMSETextStyle Class](#)

Enumerates the types of text justification.

```
enum Justification {  
    kLeft,  
    kCenter,  
    kMiddle,  
    kRight,  
    kTopLeft,  
    kTopCenter,  
    kTopRight,  
    kMiddleLeft,  
    kMiddleCenter,  
    kMiddleRight,  
    kBottomLeft,  
    kBottomCenter,  
    kBottomRight,  
    kNone  
};  
File
```

DmSETextStyle.h

Parameters	Description
kLeft	Left.
kCenter	Center.
kMiddle	Middle.
kRight	Right.
kTopLeft	Top left.
kTopCenter	Top center.
kTopRight	Top right.
kMiddleLeft	Middle left.
kMiddleCenter	Middle center.
kMiddleRight	Middle right.
kBottomLeft	Bottom left.
kBottomCenter	Bottom center.

kBottomRight            Bottom right.

kNone                    None.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
AcMapDMSETextStyle:: ~AcMapDMSETextStyle Destructor  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMSETextStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)  
[AcMapDMSETextStyle:: AcMapDMSETextStyle Constructor](#)  
[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Constructs an instance of this class.

```
AcMapDMSETextStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: clone Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: ColorIsExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether the color is an expression.

**bool** ColorIsExpression();

Returns

Returns true if the color is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: copyFrom Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: DeleteCookie Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input opaque memory value.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle::Dismiss Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Removes stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output cookie stored by Update().
id	Input object ID.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: dwgInFields Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: dwgOutFields Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: dxfInFields Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: dxfOutFields Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: Enable Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Enables or disables the style for the specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value.
id	Input object ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetColor Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text color.

```
Acad::ErrorStatus GetColor(  
    AcCmColor& color  
) const;
```

Parameters	Description
color	Output text color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetHeight Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text height.

```
Acad::ErrorStatus GetHeight(  
    double& dHeight  
) const;
```

Parameters	Description
dHeight	Output height.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetJustification Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text justification.

```
Acad::ErrorStatus GetJustification(  
    Justification& justification  
) const;
```

Parameters	Description
justification	Output Justificationvalue.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetLayer Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text layer id.

```
Acad::ErrorStatus GetLayer(  
    AcDbObjectId& layerId  
) const;
```

Parameters	Description
layerId	Output layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetRotation Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text rotation.

```
Acad::ErrorStatus GetRotation(  
    double& dRotation  
) const;
```

Parameters	Description
dRotation	Output rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetStyle Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the text style id.

```
Acad::ErrorStatus GetStyle(  
    AcDbObjectId& styleId  
) const;
```

Parameters	Description
styleId	Output style AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: GetStylizationEntities Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Retrieves the AcDbObjectIds of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input opaque memory value.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: HeightIsExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether the height is an expression.

**bool** HeightIsExpression();

Returns

Returns true if the height is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: JustificationIsExpression Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether justification is an expression.

**bool** JustificationIsExpression();

Returns

Returns true if justification is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: LayerIsExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether the layer is an expression.

**bool** LayerIsExpression();

Returns

Returns true if the layer is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle::RotationIsExpression Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether the rotation is an expression.

**bool** RotationIsExpression();

Returns

Returns true if the rotation is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetColor Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text color.

```
Acad::ErrorStatus SetColor(  
    AcCmColor color  
);
```

Parameters	Description
color	Input text color.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetHeight Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text height.

```
Acad::ErrorStatus SetHeight(  
    double dHeight  
);
```

Parameters	Description
dHeight	Input text height.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetJustification Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text justification.

```
Acad::ErrorStatus SetJustification(  
    Justification justification  
);
```

Parameters	Description
justification	Input Justificationvalue.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: SetLayer Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the layer by id.

```
Acad::ErrorStatus SetLayer(  
    const AcDbObjectId layerId  
);
```

Parameters	Description
layerId	Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

AcMapDMSETextStyle:: SetRotation Method

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text rotation.

```
Acad::ErrorStatus SetRotation(  
    double dRotation  
);
```

Parameters	Description
dRotation	Input rotation.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: SetStyle Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Sets the text style by id.

```
Acad::ErrorStatus SetStyle(  
    const AcDbObjectId styleId  
);
```

Parameters	Description
styleId	Input style AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: StyleIsExpression Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Determines whether the style is an expression.

**bool** StyleIsExpression();

Returns

Returns true if the style is an expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMSETextStyle Class](#), [AcMapDMSETextStyle Class](#)

[AcMapDMSETextStyle:: Update Method](#)

[AcMapDMSETextStyle Class](#) | [AcMapDMSETextStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input opaque memory value for later access by Dismiss().
id	Input object ID.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#)

AcMapDMStyle:: CachableProperties Enumeration

[AcMapDMStyle Class](#)

Enumerates ADE non-geometric DOT variables.

```
enum CachableProperties {  
    kUndefined,  
    kTag,  
    kString,  
    kBlockName,  
    kClassName,  
    kImageName,  
    kLayerName,  
    kLinetypeName,  
    kPlotStyleName,  
    kShapeName,  
    kTextStyleName,  
    kURL,  
    kTopoName,  
    kLineWeight,  
    kColorACI,  
    kTrueColor  
};  
File
```

DmDisplayStyle.h

Parameters	Description
kUndefined	For internal use only.
kTag	AcDbAttribute tag. Pass as ACHAR*.
kString	AcDbText or attribute string value. Pass as ACHAR*.
kBlockName	Block name. Pass as ACHAR*.
kClassName	Feature classification class name. Pass as ACHAR*.
kImageName	Image filename. Pass as ACHAR*.
kLayerName	Layer name. Pass as ACHAR*.
kLinetypeName	Linetype name. Pass as ACHAR*.
kPlotStyleName	Plotstyle name. Pass as ACHAR*.
kShapeName	Shape name. Pass as ACHAR*.

kTextStyleName	Text style name. Pass as ACHAR*.
kURL	kURL (exact string). Pass as ACHAR*.
kTopoName	ADE Topology name. Pass as ACHAR*.
kLineWeight	Lineweight. Pass as int.
kColorACI	ACI color. Pass as int.
kTrueColor	True color. Pass as AcCmColor.
Remarks	

Styles that modify one of the listed properties must cache the stylized version of the value at Update() time by using the appropriate [CacheStylizedPropValue\(\)](#) function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Apply Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Applies this style to a specified entity.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to apply the style to. This entity must have been opened for write.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: CacheStylizedPropValue Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized color property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    AcDbObjectId entityId,  
    CachableProperties key,  
    const AcCmColor& color  
) const;
```

Parameters	Description
entityId	Input ID of the entity.
key	Input CachablePropertieskey.
color	Input color value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: CacheStylizedPropValue Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Caches the stylized integer property value.

```
Acad::ErrorStatus CacheStylizedPropValue(  
    AcDbObjectId entityId,  
    CachableProperties key,  
    int value  
) const;
```

Parameters	Description
entityId	Input ID of the entity.
key	Input CachablePropertieskey.
value	Input integer value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle::copyFrom Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: DecrementRef Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Decrements the reference count to indicate there is one fewer client using this style. If the reference count becomes zero, the style erases itself. You do not need to call this function directly unless you are implementing an [AcMapDMStyleReference](#) class. See also [IncrementRef\(\)](#).

```
Adesk::UInt32 DecrementRef();
```

Returns

Returns number of references to this style.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: DeleteCookie Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input cookie to delete.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle::Dismiss Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Removes the stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: dwgInFields Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: dwgOutFields Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: dxfInFields Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: dxfOutFields Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Enable Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Enables or disables the style for the specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input entity ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: GetName Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Retrieves the name of this style. See also [SetName\(\)](#).

```
virtual const ACHAR* GetName() const;
```

Returns

Returns the style name, or NULL if unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: GetStylizationEntities Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Retrieves the AcDbObjectIds of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: Implementation Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Returns the implementation object.

```
virtual AcMapDMImpStyle* Implementation();
```

Returns

Returns the implementation object, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: IncrementRef Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Increments the reference count to indicate that there is one more client using this style. You do not need to call this function directly unless you are implementing an [AcMapDMStyleReference](#) class. The caller must call `DecrementRef()` when finished with this style.

```
Adesk::UInt32 IncrementRef();
```

Returns

Returns the number of references to this style.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: IsMultiplyReferenced Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Determines whether this style has multiple references.

**bool** IsMultiplyReferenced() **const**;

Returns

Returns true if multiple references exist.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: OnMapProjectInitialized Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Invoked when the an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pProject  
);
```

Parameters	Description
pProject	Input AutoCAD Map project.

Returns

Returns true if successful.

## Remarks

The style can perform any project-related initialization at this time, and should return true if successful, or false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: OnObjectAppended Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Invoked after a style has been appended to the database, allowing the appended style to do any necessary post-append initialization.

```
virtual void OnObjectAppended();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Project Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Retrieves the AutoCAD Map project.

```
AcMapProject* Project();
```

Returns

Returns the AutoCAD Map project, or NULL if unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: subClose Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subClose();
```

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: subErase Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

[AcMapDMStyle:: UnApply Method](#)

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Removes the style from a specified entity.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: Update Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyle Class](#), [AcMapDMStyle Class](#)

AcMapDMStyle:: wblockClone Method

[AcMapDMStyle Class](#) | [AcMapDMStyle Class](#)

Clones this style. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus wblockClone(  
    AcRxObject* pOwnerObject,  
    AcDbObject*& pClonedObject,  
    AcDbIdMapping& idMap,  
    Adesk::Boolean isPrimary = true  
) const;
```

Parameters	Description
pOwnerObject	Input object to append the clone to.
pClonedObject	Returned cloned object, or NULL if not cloned.
idMap	Input current object map ID.
isPrimary	Input true if this object is primary, or false if it is owned.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)  
AcMapDMStyleCategory:: ~AcMapDMStyleCategory Destructor  
[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMStyleCategory(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)  
AcMapDMStyleCategory:: AcMapDMStyleCategory Constructor  
[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Constructs an instance of this class.

```
AcMapDMStyleCategory(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Append Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Adds a style to this category.

```
Acad::ErrorStatus Append(  
    AcDbObjectId& Id,  
    AcMapDMStyle* pStyle  
);
```

Parameters	Description
Id	Output ID of the style.
pStyle	Input AcMapDMStyleobject.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: AppendStyleCopy Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Copies a style and appends it to this category.

```
Acad::ErrorStatus AppendStyleCopy(  
    AcDbObjectId& newId,  
    AcDbObjectId& existingId  
);
```

Parameters	Description
newId	Output ID of the new style.
existingId	Input style ID.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: AppendStyleCopy Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Copies multiple styles and appends them to this category.

```
Acad::ErrorStatus AppendStyleCopy(  
    AcDbObjectIdArray& newIdArray,  
    AcDbObjectIdArray& existingIdArray  
);
```

Parameters	Description
newIdArray	Output array of IDs of the new styles.
existingIdArray	Input array of IDs of existing styles.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory::audit Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters

Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: dwgInFields Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: dwgOutFields Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: dxfInFields Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: dxfOutFields Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: erased Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Lets this object listen to erase-notifications from items that it owns. See also erased() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual void erased(  
    const AcDbObject* dbObj,  
    Adesk::Boolean bErasing = true  
);
```

Parameters	Description
dbObj	Input object that was erased.
bErasing	Input true to listen as an erase is happening. The default value is true.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Find Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Finds a specified style within this category.

```
bool Find(  
    Adesk::UInt16& nIndex,  
    const AcDbObjectId& Id  
) const;
```

Parameters	Description
nIndex	Output index of the style, if found.
Id	Input ID of the style to find.

Returns

Returns true if style is found; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: GetAt Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Retrieves the ID of a style at a specified position.

```
Acad::ErrorStatus GetAt(  
    AcDbObjectId& Id,  
    Adesk::UInt16 nIndex  
) const;
```

Parameters	Description
Id	Output ID of the style.
nIndex	Input position of the style.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: GetAt Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Retrieves a style at a specified position, opened in a specified mode.

```
Acad::ErrorStatus GetAt(  
    AcMapDMStyle*& pStyle,  
    Adesk::UInt16 nIndex,  
    AcDb::OpenMode mode  
) const;
```

Parameters	Description
pStyle	Output AcMapDMStyleobject.
nIndex	Input position of the style.
mode	Input mode in which the style is to be opened.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: GetAt Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Retrieves a style, opened in a specified mode.

```
Acad::ErrorStatus GetAt(  
    AcMapDMStyle*& pStyle,  
    const AcDbObjectId& Id,  
    AcDb::OpenMode mode  
) const;
```

Parameters	Description
pStyle	Output AcMapDMStyleobject.
Id	Input ID of the style.
mode	Input mode in which the style is to be opened.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory:: GetName Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Retrieves the name of this category. See also [SetName\(\)](#).

```
virtual const ACHAR* GetName() const;
```

Returns

Returns the category name if successful; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Has Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Determines whether this category has a specified style.

```
bool Has(  
    const AcDbObjectId& objId  
) const;
```

Parameters	Description
objId	Input ID of the style.

Returns

Returns true if the category has the specified style.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory:: Implementation Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Returns the implementation object.

```
AcMapDMImpStyleCategory* Implementation();
```

Returns

Returns the implementation object, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory::InsertAt Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Inserts a style at a specified position.

```
Acad::ErrorStatus InsertAt(  
    AcDbObjectId& Id,  
    AcMapDMStyle* pStyle,  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
Id	Output ID of the style.
pStyle	Input AcMapDMStyleobject.
nIndex	Input position.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory::InsertStyleCopyAt Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Inserts a copy of a style at a specified position.

```
Acad::ErrorStatus InsertStyleCopyAt(  
    AcDbObjectId& newId,  
    AcDbObjectId& existingId,  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
newId	Output ID of the inserted style.
existingId	Input ID of the existing style.
nIndex	Input position of the inserted style.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)  
AcMapDMStyleCategory::InsertStyleCopyAt Method  
[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Inserts multiple styles at a specified position.

```
Acad::ErrorStatus InsertStyleCopyAt(  
    AcDbObjectIdArray& newIdArray,  
    AcDbObjectIdArray& existingIdArray,  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
newIdArray	Output array of IDs of the inserted styles.
existingIdArray	Input array of IDs of existing styles.
nIndex	Input position of the inserted styles.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Move Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Moves a style to a new position.

```
Acad::ErrorStatus Move(  
    Adesk::UInt16 nOldIndex,  
    Adesk::UInt16 nNewIndex  
);
```

Parameters	Description
nOldIndex	Input old position.
nNewIndex	Input new position.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Move Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Moves a style to a new position.

```
Acad::ErrorStatus Move(  
    const AcDbObjectId& Id,  
    Adesk::UInt16 nNewIndex  
);
```

Parameters	Description
Id	Input style ID.
nNewIndex	Input new position.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory:: NumStyles Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Counts the number of styles within this category.

```
Adesk::UInt16 NumStyles() const;
```

Returns

Returns the number of styles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: Remove Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Removes a style.

```
Acad::ErrorStatus Remove(  
    const AcDbObjectId& Id  
);
```

Parameters

Description

Id

Input ID of the style to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory:: RemoveAt Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Removes a style at a specified position.

```
Acad::ErrorStatus RemoveAt(  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
nIndex	Input position of the style to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

[AcMapDMStyleCategory:: subClose Method](#)

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subClose();
```

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleCategory Class](#), [AcMapDMStyleCategory Class](#)

AcMapDMStyleCategory:: subErase Method

[AcMapDMStyleCategory Class](#) | [AcMapDMStyleCategory Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

Parameters	Description
erasing	Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary::~~AcMapDMStyleLibrary Destructor  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMStyleLibrary(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: AcMapDMStyleLibrary Constructor  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Constructs an instance of this class.

```
AcMapDMStyleLibrary(  
    void  
);
```

Parameters	Description
void	Void.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: Append Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Adds a style category to this library.

```
Acad::ErrorStatus Append(  
    AcDbObjectId& Id,  
    AcMapDMStyleCategory* pStyleCategory  
);
```

Parameters	Description
Id	Output ID of the category.
pStyleCategory	Input <a href="#">AcMapDMStyleCategory</a> object.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: audit Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters	Description
pAuditInfo	The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

## Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: dwgInFields Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

AcMapDMStyleLibrary:: dwgOutFields Method

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: dxfInFields Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

[AcMapDMStyleLibrary:: dxfOutFields Method](#)

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

[AcMapDMStyleLibrary:: erased Method](#)

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Lets this object listen to erase-notifications from items that it owns. See also `erased()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual void erased(  
    const AcDbObject* dbObj,  
    Adesk::Boolean bErasing = true  
);
```

Parameters	Description
dbObj	Input object that was erased.
bErasing	Input true to listen as an erase is happening. The default value is true.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

[AcMapDMStyleLibrary:: GetAt Method](#)

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Retrieves a category at a specified position.

```
Acad::ErrorStatus GetAt(  
    AcDbObjectId& Id,  
    Adesk::UInt16 nIndex  
) const;
```

Parameters	Description
Id	Output ID of the category.
nIndex	Input position of the category.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

AcMapDMStyleLibrary:: GetAt Method

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Retrieves a category at a specified position, opened in a specified mode.

```
Acad::ErrorStatus GetAt(  
    AcMapDMStyleCategory*& pStyleCategory,  
    Adesk::UInt16 nIndex,  
    AcDb::OpenMode mode  
) const;
```

Parameters	Description
pStyleCategory	Output <a href="#">AcMapDMStyleCategory</a> object.
nIndex	Input position of the category.
mode	Input mode in which the category is to be opened.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

AcMapDMStyleLibrary:: GetAt Method

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Retrieves a category, opened in a specified mode.

```
Acad::ErrorStatus GetAt(  
    AcMapDMStyleCategory*& pStyleCategory,  
    const AcDbObjectId& Id,  
    AcDb::OpenMode mode  
) const;
```

Parameters	Description
pStyleCategory	Output <a href="#">AcMapDMStyleCategory</a> object.
Id	Input ID of the category.
mode	Input mode in which the category is to be opened.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: Has Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Determines whether a specific category exists in this library.

```
bool Has(  
    const AcDbObjectId& Id  
) const;
```

Parameters	Description
Id	Input ID of the category.

Returns

Returns true if the category exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: Implementation Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Returns the implementation object.

```
AcMapDMImpStyleLibrary* Implementation();
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary::InsertAt Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Inserts a category at a specified position.

```
Acad::ErrorStatus InsertAt(  
    AcDbObjectId& Id,  
    AcMapDMStyleCategory* pStyleCategory,  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
Id	Output ID of the category.
pStyleCategory	Input <a href="#">AcMapDMStyleCategory</a> object.
nIndex	Input position.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
[AcMapDMStyleLibrary:: Move Method](#)  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Moves a category from one position to another in this library.

```
Acad::ErrorStatus Move(  
    Adesk::UInt16 nOldIndex,  
    Adesk::UInt16 nNewIndex  
);
```

Parameters	Description
nOldIndex	Input old position.
nNewIndex	Input new position.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)

AcMapDMStyleLibrary:: Move Method

[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Moves a category with a specified ID to another position in this library.

```
Acad::ErrorStatus Move(  
    const AcDbObjectId& Id,  
    Adesk::UInt16 nNewIndex  
);
```

Parameters	Description
Id	Input ID of the category.
nNewIndex	Input new position.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: NumCategories Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Counts the number of categories within this library.

```
Adesk::UInt16 NumCategories() const;
```

Returns

Returns the number of categories.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: Remove Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Removes a category from this library.

```
Acad::ErrorStatus Remove(  
    const AcDbObjectId& Id  
);
```

Parameters	Description
Id	Input ID of the category to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleLibrary Class](#), [AcMapDMStyleLibrary Class](#)  
AcMapDMStyleLibrary:: RemoveAt Method  
[AcMapDMStyleLibrary Class](#) | [AcMapDMStyleLibrary Class](#)

Removes a category at a specified position from this library.

```
Acad::ErrorStatus RemoveAt(  
    Adesk::UInt16 nIndex  
);
```

Parameters	Description
nIndex	Input position of the category to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)  
AcMapDMStyleReference:: ~AcMapDMStyleReference Destructor  
[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMStyleReference( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: AcMapDMStyleReference Constructor

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Constructs an instance of this class.

```
AcMapDMStyleReference();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference::audit Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

This function is called by AutoCAD when the AUDIT command is executed.

```
virtual Acad::ErrorStatus audit(  
    AcDbAuditInfo* pAuditInfo  
);
```

Parameters

Description

pAuditInfo

The AcDbAuditInfo object pointed to by pAuditInfo contains member functions that are used to determine what to do and also to report the results of the audit operation on the object.

Returns

Returns Acad::eOk if successful, Acad::eFixedAllErrors if all errors were corrected, Acad::eLeftErrorsUnfixed if there were errors that were left unfixed; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

[AcMapDMStyleReference:: dwgInFields Method](#)

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
<code>pFiler</code>	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

[AcMapDMStyleReference:: dwgOutFields Method](#)

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
<code>pFiler</code>	Input filer to use to write the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

[AcMapDMStyleReference:: dxfInFields Method](#)

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: dxfOutFields Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)  
AcMapDMStyleReference:: Implementation Method  
[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Returns the implementation object.

```
AcMapDMImpStyleReference* Implementation() const;  
Returns
```

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: IsEnabled Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Determines whether this reference is enabled. See also SetEnabled().

```
bool IsEnabled() const;
```

Returns

Returns true if the reference is enabled, or false if it is disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference::IsUniqueReference Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Determines whether this reference is the only reference to the style that this reference refers to. See also `MakeUniqueReference()`.

```
bool IsUniqueReference() const;
```

Returns

Returns true if this is the only reference; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference::MakeUniqueReference Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Determines whether this reference is a unique reference and, if not, clones the referenced style and resets the style ID to the new copy. See also `IsUniqueReference()`.

```
Acad::ErrorStatus MakeUniqueReference();
```

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: SetEnabled Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Enables or disables this reference. See also IsEnabled().

```
Acad::ErrorStatus SetEnabled(  
    bool bNewStatus  
);
```

Parameters	Description
bNewStatus	Input new enabled status.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: StyleId Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Retrieves the ID of the style that this reference refers to.

```
AcDbObjectId StyleId() const;
```

Returns

Returns the style ID if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

[AcMapDMStyleReference:: subClose Method](#)

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subClose();
```

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

AcMapDMStyleReference:: subErase Method

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStyleReference Class](#), [AcMapDMStyleReference Class](#)

[AcMapDMStyleReference:: wblockClone Method](#)

[AcMapDMStyleReference Class](#) | [AcMapDMStyleReference Class](#)

Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object pOwnerObject. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus wblockClone(  
    AcRxObject* pOwnerObject,  
    AcDbObject*& pClonedObject,  
    AcDbIdMapping& idMap,  
    Adesk::Boolean isPrimary = true  
) const;
```

Parameters	Description
pOwnerObject	Input object to append the clones to.
pClonedObject	Output the cloned object, or NULL if not cloned.
idMap	Input current ID map.
isPrimary	Input true if this object is primary, or false if it is owned.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: Apply Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Applies the style to an entity.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to apply the style to.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
AcMapDMStylizationEntityStyle:: CacheStylizationEntity Method  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Informs the display manager that a particular AcDbEntity was created as a result of applying a StylizationEntityStyle.

```
Acad::ErrorStatus CacheStylizationEntity(  
    const AcDbObjectId styleEntId  
) const;
```

Parameters	Description
styleEntId	Input ID of the stylization entity.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

A derived class must call this function when it creates a new stylization entity.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle::Dismiss Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Removes stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: dwgInFields Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: dwgOutFields Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: dxfInFields Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: dxfOutFields Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)

[AcMapDMStylizationEntityStyle:: Enable Method](#)

[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Enables or disables the style for the specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)

[AcMapDMStylizationEntityStyle::](#)

[ExcludeStylizationEntityFromHatchStyle Method](#)

[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Creates an island around the stylization entity.

```
Acad::ErrorStatus ExcludeStylizationEntityFromHatchStyle(  
    const AcDbObjectId id,  
    const AcDbObjectId styleEntId  
) const;
```

Parameters	Description
id	Input ID of the target entity.
styleEntId	Input ID of the stylization entity.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

A derived class must call this function if it creates a stylization entity, and wants the AcMapDMSEHatchStyle to attempt to create an island around the stylization entity, should a stylization hatch happen to be applied to the same target entity. In this release of AutoCAD Map, islanding works for only AcDbText entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: GetStylizationEntities Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Retrieves the AcDbObjectIds of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array IDs of of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input opaque memory value.
flag	Reserved. For internal use only.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: UnApply Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)

[AcMapDMStylizationEntityStyle::](#)

[UnExcludeStylizationEntityFromHatchStyle Method](#)

[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

A derived class must call this function to remove a stylization entity from association with a target entity, for the purposes of AcMapDMSEHatchStyle islanding.

```
Acad::ErrorStatus UnExcludeStylizationEntityFromHatchStyle(  
    const AcDbObjectId id,  
    const AcDbObjectId styleEntId  
) const;
```

Parameters	Description
id	Input ID of the target entity.
styleEntId	Input ID of the stylization entity.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This should be done by the Style at dismiss time if the Style may reuse the same stylization entity against a different target entity. In this release of AutoCAD Map, islanding works for only AcDbText entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMStylizationEntityStyle Class](#), [AcMapDMStylizationEntityStyle Class](#)  
[AcMapDMStylizationEntityStyle:: Update Method](#)  
[AcMapDMStylizationEntityStyle Class](#) | [AcMapDMStylizationEntityStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
AcMapDMThematicStyle::~~AcMapDMThematicStyle Destructor  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
[AcMapDMThematicStyle:: AcMapDMThematicStyle Constructor](#)  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Constructs an instance of this class.

```
AcMapDMThematicStyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Apply Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Applies the style to an entity.

```
virtual Acad::ErrorStatus Apply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity to apply the style to.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: AutoRecalculateRanges Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether automatic recalculation of ranges is turned on or off.

```
virtual bool AutoRecalculateRanges() const;
```

Returns

Returns true if automatic recalculation is turned on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: DataSourceExpression Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the expression used in theming.

```
virtual const ACHAR * DataSourceExpression() const;
```

Returns

Returns the expression, or NULL if the expression is not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: DeleteCookie Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Deletes the style's cookie during database destruction.

```
virtual void DeleteCookie(  
    void*& pCookie  
);
```

Parameters	Description
pCookie	Input cookie to delete.
Returns	

Returns nothing.

## Remarks

Because the database is in the midst of destruction, do not attempt any additional work by using the cookie at this point, other than its deletion.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle::Dismiss Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Removes stylization information that is used during regeneration.

```
virtual Acad::ErrorStatus Dismiss(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: dwgInFields Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: dwgOutFields Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: dxfInFields Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: dxfOutFields Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Enable Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Enables or disables the style for a specified entity.

```
virtual Acad::ErrorStatus Enable(  
    void*& pCookie,  
    const AcDbObjectId id,  
    bool bEnable,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
bEnable	Input true to enable the style.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: GetStylizationEntities Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the IDs of any stylization entities created during an Update().

```
virtual Acad::ErrorStatus GetStylizationEntities(  
    AcDbObjectIdArray& output,  
    const AcDbObjectId id,  
    void* pCookie,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
output	Output array of IDs of stylization entities.
id	Target entity against which the style was applied.
pCookie	Input opaque memory value.
flag	Reserved. For internal use only.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: GetThematicTable Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the thematic table, opened in a specified mode.

```
virtual Acad::ErrorStatus GetThematicTable(  
    AcMapDMThematicTable*& pTable,  
    Acad::OpenMode openMode  
);
```

Parameters	Description
pTable	Output <a href="#">AcMapDMThematicTable</a> thematic table.
openMode	Input mode in which the table is to be opened.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: GroupCount Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Counts the number of groups for thematic ranges.

```
virtual int GroupCount() const;
```

Returns

Returns the number of groups.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: GroupingAlgorithmName Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the name of the grouping algorithm.

```
virtual const ACHAR * GroupingAlgorithmName() const;
```

Returns

Returns the name of the grouping algorithm.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: GroupRoundingMethod Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the name of the rounding method used.

```
virtual const ACHAR * GroupRoundingMethod() const;
```

Returns

Returns the name of the rounding method used.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterAnnotationEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether annotation alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterAnnotationEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterBlockEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether block alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterBlockEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterColorEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether color alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterColorEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterHatchEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether hatch alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterHatchEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: IsAlterLineStyleEnabled Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether linestyle alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterLineStyleEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: IsAlterLinetypeEnabled Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether linetype alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterLinetypeEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterLineweightEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether lineweight alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterLineweightEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterPlotstyleEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether plotstyle alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterPlotstyleEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsAlterTextEnabled Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether text alteration is set to modify a thematic stylization query result.

```
virtual bool IsAlterTextEnabled() const;
```

Returns

Returns true if alteration is on.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsScaleColorRamp Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether the scale-color-ramp flag is set.

```
virtual bool IsScaleColorRamp() const;
```

Returns

Returns true if the flag is set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: IsScaleHatchRamp Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether the scale-hatch-ramp flag is set.

```
virtual bool IsScaleHatchRamp() const;
```

Returns

Returns true if the flag is set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsScaleLineStyleRamp Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether the scale-linestyle-ramp flag is set.

```
virtual bool IsScaleLineStyleRamp() const;
```

Returns

Returns true if the flag is set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: IsScaleRampToFit Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether the scale-ramp-to-fit flag is set.

```
virtual bool IsScaleRampToFit() const;
```

Returns

Returns true if the flag is set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: IsSourceDataNumeric Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether the data source is numeric.

```
virtual bool IsSourceDataNumeric() const;
```

Returns

Returns true if the data source is numeric.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: LastColorRampName Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the name of the last-used color ramp.

```
virtual const ACHAR * LastColorRampName() const;
```

Returns

Returns the ramp name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: LastHatchRampName Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the name of the last-used hatch ramp.

```
virtual const ACHAR * LastHatchRampName() const;
```

Returns

Returns the ramp name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: LastLineStyleRampName Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the name of the last-used linestyle ramp.

```
virtual const ACHAR * LastLineStyleRampName() const;
```

Returns

Returns the ramp name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: LinetypeScale Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the linetype scale.

```
virtual double LinetypeScale() const;
```

Returns

Returns the linetype scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

AcMapDMThematicStyle:: NormalizationExpression Method

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the normalization expression.

```
virtual const ACHAR * NormalizationExpression() const;
```

Returns

Returns the expression, or NULL if the expression is not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: OnObjectAppended Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Invoked when an object is assigned to the database.

```
virtual void OnObjectAppended();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: RangeTableComparisonOperator Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the range-table comparison operator.

```
virtual AcMap::ERangeOperator RangeTableComparisonOperator() const;
```

Returns

Returns an range-operator enum value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterAnnotation Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets annotation alteration.

```
virtual Acad::ErrorStatus SetAlterAnnotation(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterBlock Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets block alteration.

```
virtual Acad::ErrorStatus SetAlterBlock(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterColor Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets color alteration.

```
virtual Acad::ErrorStatus SetAlterColor(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterHatch Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets hatch alteration.

```
virtual Acad::ErrorStatus SetAlterHatch(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterLinestyle Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets linestyle alteration.

```
virtual Acad::ErrorStatus SetAlterLinestyle(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterLinetype Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets linetype alteration.

```
virtual Acad::ErrorStatus SetAlterLinetype(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterLineweight Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets lineweight alteration.

```
virtual Acad::ErrorStatus SetAlterLineweight(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterPlotstyle Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets plotstyle alteration.

```
virtual Acad::ErrorStatus SetAlterPlotstyle(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetAlterText Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets text alteration.

```
virtual Acad::ErrorStatus SetAlterText(  
    bool bEnable = true  
);
```

Parameters

Description

bEnable

Input true to turn alteration on.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
[AcMapDMThematicStyle:: SetAutoRecalculateRanges Method](#)  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Turns automatic recalculation on or off.

```
virtual Acad::ErrorStatus SetAutoRecalculateRanges(  
    bool bRecalculateRanges  
);
```

Parameters	Description
------------	-------------

bRecalculateRanges	Input true to turn automatic recalculation on.
--------------------	--

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetGroupCount Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the number of groups for thematic ranges.

```
virtual Acad::ErrorStatus SetGroupCount(  
    int groupCount  
);
```

Parameters	Description
groupCount	Input number of groups.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetLinetypeScale Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the linetype scale.

```
virtual Acad::ErrorStatus SetLinetypeScale(  
    double newLinetypeScale  
);
```

Parameters	Description
newLinetypeScale	Input linetype scale.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetRangeTableComparisonOperator Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the range-table comparison operator.

```
virtual Acad::ErrorStatus SetRangeTableComparisonOperator(  
    AcMap::ERangeOperator newRangeOperator  
);
```

Parameters	Description
newRangeOperator	Input range-operator value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetScaleColorRamp Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the scale-color-ramp flag.

```
virtual Acad::ErrorStatus SetScaleColorRamp(  
    bool bIsScaleColorRamp  
);
```

Parameters	Description
bIsScaleColorRamp	Input true to set the flag.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetScaleHatchRamp Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the scale-hatch-ramp flag.

```
virtual Acad::ErrorStatus SetScaleHatchRamp(  
    bool bIsScaleHatchRamp  
);
```

Parameters	Description
bIsScaleHatchRamp	Input true to set the flag.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetScaleLineStyleRamp Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the scale-linestyle-ramp flag.

```
virtual Acad::ErrorStatus SetScaleLineStyleRamp(  
    bool bIsScaleLineStyleRamp  
);
```

Parameters	Description
------------	-------------

bIsScaleLineStyleRamp	Input true to set the flag.
-----------------------	-----------------------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetScaleRampToFit Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the scale-ramp-to-fit flag.

```
virtual Acad::ErrorStatus SetScaleRampToFit(  
    bool bIsScaleRampToFit  
);
```

Parameters	Description
bIsScaleRampToFit	Input true to set the flag.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: SetSourceDataNumeric Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the data source of the thematic stylization to be numeric.

```
virtual Acad::ErrorStatus SetSourceDataNumeric(  
    bool bSourceDataIsNumeric  
);
```

Parameters

Description

bSourceDataIsNumeric Input true to make the data source numeric.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)  
[AcMapDMThematicStyle:: SetUseThousandsSeparators Method](#)  
[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Sets the use of thousands separators.

```
virtual Acad::ErrorStatus SetUseThousandsSeparators(  
    bool bUseThousandsSeparators  
);
```

Parameters

Description

bUseThousandsSeparators Input true to use thousands separators.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: subErase Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Passed-in copy of the erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: UnApply Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus UnApply(  
    void*& pCookie,  
    AcDbEntity* pEnt,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Input memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
pEnt	Input entity ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The caller can retrieve this style's cookie, stored previously by another stylization function.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: Update Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Builds stylization information for later regeneration.

```
virtual Acad::ErrorStatus Update(  
    void*& pCookie,  
    const AcDbObjectId id,  
    Adesk::UInt32 flag = 0  
);
```

Parameters	Description
pCookie	Output memory value used by styles to transfer cached data across time from one stylization method to another. Cookie memory is considered to be opaque for existing styles, and should not be modified by the caller.
id	Input object ID.
flag	Reserved. For internal use only.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Derived classes can perform any one-time Update()-time actions here. At Dismiss() time, the implementation can do any required work to clean up storage created at Update() time.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: UseThousandsSeparators Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Determines whether thousands separators are used.

```
virtual bool UseThousandsSeparators() const;
```

Returns

Returns true if thousand separators are used.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicStyle Class](#), [AcMapDMThematicStyle Class](#)

[AcMapDMThematicStyle:: ValuesToIgnore Method](#)

[AcMapDMThematicStyle Class](#) | [AcMapDMThematicStyle Class](#)

Retrieves the values-to-ignore expression.

```
virtual const ACHAR * ValuesToIgnore() const;
```

Returns

Returns the expression, or NULL if the expression is not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: ~AcMapDMThematicTable Destructor](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTable();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: AcMapDMThematicTable Constructor](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTable();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: AppendRow Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Appends a specified row to the thematic table.

```
virtual Acad::ErrorStatus AppendRow(  
    AcMapDMThematicTableRow& row  
);
```

Parameters

Description

row

Input [AcMapDMThematicTableRow](#) object.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: Clear Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Empties all contents of the thematic table.

```
virtual Acad::ErrorStatus Clear();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: dwgInFields Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Lets this object read its data. See also `dwgInFields()` in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

`pFiler`

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: dwgOutFields Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Lets this object write its data. See also `dwgOutFields()` in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

`pFiler`

Input filer to use to write the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: dxfInFields Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: dxfOutFields Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves an annotation cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemAnnotation*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemAnnotation</a> value.
row	Input row number.
openMode	Input open mode.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a block cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemBlock*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemBlock</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a color cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemColor*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemColor</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: GetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a data cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemDataValue*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemDataValue</a> value.
row	Input row number.
openMode	Input open mode.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a hatch cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemHatch*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemHatch</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a legend-text cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemLegendText* & cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemLegendText</a> value.
row	Input row number.
openMode	Input open mode.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a linestyle cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemLinestyle*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemLinestyle</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a linetype cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemLinetype*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemLinetype</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a lineweight cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemLineweight* & cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemLineweight</a> value.
row	Input row number.
openMode	Input open mode.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a plotstyle cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemPlotstyle*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemPlotstyle</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves a text cell item.

```
virtual Acad::ErrorStatus GetCellAt(  
    AcMapDMThematicTableItemText*& cellItem,  
    Adesk::UInt32 row,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
cellItem	Output <a href="#">AcMapDMThematicTableItemText</a> value.
row	Input row number.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: GetRowAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Opens and retrieves an entire row.

```
virtual Acad::ErrorStatus GetRowAt(  
    AcMapDMThematicTableRow& row,  
    Adesk::UInt32 rowIndex,  
    AcDb::OpenMode openMode  
);
```

Parameters	Description
row	Output <a href="#">AcMapDMThematicTableRow</a> object.
rowIndex	Input row index.
openMode	Input open mode.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: Implementation Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTable* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the annotation style for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemAnnotation& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemAnnotation</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: SetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the block for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemBlock& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemBlock</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: SetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the item color for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemColor& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemColor</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the value for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemDataValue& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemDataValue</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function fails if cellItem already is appended to the database.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: SetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the hatch for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemHatch& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemHatch</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: SetCellAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the legend text for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemLegendText& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemLegendText</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the linestyle for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemLinestyle& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemLinestyle</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the linetype for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemLinetype& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemLinetype</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the linewidth for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemLineweight& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemLineweight</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the plotstyle for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemPlotstyle& cellItem  
);
```

Parameters	Description
row	Input row number.
cellItem	Input <a href="#">AcMapDMThematicTableItemPlotstyle</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

AcMapDMThematicTable:: SetCellAt Method

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the text for a specified cell.

```
virtual Acad::ErrorStatus SetCellAt(  
    Adesk::UInt32 row,  
    AcMapDMThematicTableItemText& cellItem  
);
```

Parameters

Description

row

Input row number.

cellItem

Input [AcMapDMThematicTableItemText](#) value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTable Class](#), [AcMapDMThematicTable Class](#)

[AcMapDMThematicTable:: SetRowAt Method](#)

[AcMapDMThematicTable Class](#) | [AcMapDMThematicTable Class](#)

Sets the row at a specified position.

```
virtual Acad::ErrorStatus SetRowAt(  
    Adesk::UInt32 rowIndex,  
    AcMapDMThematicTableRow& row  
);
```

Parameters	Description
rowIndex	Input row position.
row	Input <a href="#">AcMapDMThematicTableRow</a> object.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

~AcMapDMThematicTableItemAnnotation Destructor

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMThematicTableItemAnnotation\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

AcMapDMThematicTableItemAnnotation Constructor

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemAnnotation();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

AcMapDMThematicTableItemAnnotation Constructor

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemAnnotation(  
    const AcMapDMThematicTableItemAnnotation& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearColorOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the color override.

```
virtual Acad::ErrorStatus ClearColorOverride();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearLayerOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the layer override.

**virtual** Acad::ErrorStatus ClearLayerOverride();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearLinetypeOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the linetype override.

**virtual** Acad::ErrorStatus ClearLinetypeOverride();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearLineweightOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the lineweight override.

**virtual** Acad::ErrorStatus ClearLineweightOverride();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearRotationOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the rotation override.

```
virtual Acad::ErrorStatus ClearRotationOverride();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ClearScaleOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Disables the scale override.

**virtual** Acad::ErrorStatus ClearScaleOverride();

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ColorOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the color override.

```
virtual AcCmColor ColorOverride() const;
```

Returns

Returns the color override if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ColorOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the color override expression.

```
virtual const ACHAR* ColorOverrideExpression() const;
```

Returns

Returns the color override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: dwgInFields Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: dwgOutFields Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: dxfInFields Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: dxfOutFields Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: Implementation Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Returns the implementation object.

**virtual** AcMapDMImpThematicTableItemAnnotation\* Implementation() **cons**

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: IsColorOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the color override.

```
virtual bool IsColorOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: IsLayerOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the layer override.

```
virtual bool IsLayerOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: IsLinetypeOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the linetype override.

```
virtual bool IsLinetypeOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

IsLineweightOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the lineweight override.

```
virtual bool IsLineweightOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: IsRotationOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the rotation override.

```
virtual bool IsRotationOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: IsScaleOverrideEnabled Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Indicates the enable state of the scale override.

```
virtual bool IsScaleOverrideEnabled() const;
```

Returns

Returns true if enabled, false if disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: LayerOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the layer override.

```
virtual AcDbObjectId LayerOverride() const;
```

Returns

Returns the layer override if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: LayerOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the layer override expression.

```
virtual const ACHAR* LayerOverrideExpression() const;
```

Returns

Returns the layer override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: LinetypeOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the linetype override.

```
virtual AcDbObjectId LinetypeOverride() const;
```

Returns

Returns the linetype override if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

LinetypeOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the linetype override expression.

```
virtual const ACHAR* LinetypeOverrideExpression() const;
```

Returns

Returns the linetype override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: LineweightOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the lineweight override.

```
virtual AcDb::Lineweight LineweightOverride() const;
```

Returns

Returns the lineweight override.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

LineweightOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the lineweight override expression.

```
virtual const ACHAR* LineweightOverrideExpression() const;
```

Returns

Returns the lineweight override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

LocationOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the annotation location (insertion point) override expression.

```
virtual const ACHAR * LocationOverrideExpression() const;
```

Returns

Returns the annotation location override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: RotationOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the rotation override.

```
virtual double RotationOverride() const;
```

Returns

Returns the rotation override.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation::

RotationOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the annotation rotation override expression.

```
virtual const ACHAR * RotationOverrideExpression() const;
```

Returns

Returns the annotation rotation override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ScaleOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the scale override.

```
virtual double ScaleOverride() const;
```

Returns

Returns the scale override.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: ScaleOverrideExpression Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the annotation scale override expression.

```
virtual const ACHAR * ScaleOverrideExpression() const;
```

Returns

Returns the annotation scale override expression if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetColorOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the color override value.

```
virtual Acad::ErrorStatus SetColorOverride(  
    const AcCmColor colorOverride  
);
```

Parameters	Description
colorOverride	Input color override value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetLayerOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the layer override value.

```
virtual Acad::ErrorStatus SetLayerOverride(  
    const AcDbObjectId idLayerOverride  
);
```

Parameters

Description

idLayerOverride

Input layer override AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetLinetypeOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the linetype override value.

```
virtual Acad::ErrorStatus SetLinetypeOverride(  
    const AcDbObjectId idLinetypeOverride  
);
```

Parameters

Description

idLinetypeOverride

Input linetype override AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetLineweightOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the lineweight override value.

```
virtual Acad::ErrorStatus SetLineweightOverride(  
    AcDb::Lineweight lineweightOverride  
);
```

Parameters	Description
lineweightOverride	Input lineweight.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetRotationOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the rotation override value.

```
virtual Acad::ErrorStatus SetRotationOverride(  
    double rotationOverride  
);
```

Parameters	Description
rotationOverride	Input rotation override value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetScaleOverride Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the scale override value.

```
virtual Acad::ErrorStatus SetScaleOverride(  
    double scaleOverride  
);
```

Parameters

Description

scaleOverride

Input scale override value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: SetTemplateId Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Sets the template name for this annotation alteration.

```
virtual Acad::ErrorStatus SetTemplateId(  
    AcDbObjectId templateId  
);
```

Parameters

Description

templateId

Input object ID of the template for this annotation alteration.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemAnnotation Class](#),

[AcMapDMThematicTableItemAnnotation Class](#)

AcMapDMThematicTableItemAnnotation:: TemplateId Method

[AcMapDMThematicTableItemAnnotation Class](#) |

[AcMapDMThematicTableItemAnnotation Class](#)

Retrieves the annotation template Id for this alteration.

```
virtual AcDbObjectId TemplateId() const;
```

Returns

Returns the object ID of the annotation template if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock::~~AcMapDMThematicTableItemBlock  
Destructor

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemBlock();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock::

AcMapDMThematicTableItemBlock Constructor

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemBlock();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock::

AcMapDMThematicTableItemBlock Constructor

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemBlock(  
    const AcMapDMThematicTableItemBlock& other  
);
```

Parameters

Description

other

Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: Angle Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the angle.

```
virtual int Angle() const;
```

Returns

Returns the angle.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: BlockId Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the block AcDbObjectId.

```
AcDbObjectId BlockId() const;
```

Returns

Returns the block AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: BlockName Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the block name.

```
virtual const ACHAR* BlockName() const;
```

Returns

Returns the block name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: dwgInFields Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: dwgOutFields Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: dxfInFields Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: dxfOutFields Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: Implementation Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemBlock* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: LayerId Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the layer AcDbObjectId.

```
AcDbObjectId LayerId() const;
```

Returns

Returns the layer AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: LayerName Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the layer name.

```
virtual const ACHAR* LayerName() const;
```

Returns

Returns the layer name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: Scale Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Retrieves the scale.

```
virtual double Scale() const;
```

Returns

Returns scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetAngle Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the angle.

```
virtual Acad::ErrorStatus SetAngle(  
    int angle  
);
```

Parameters	Description
------------	-------------

angle	Input angle.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetBlockId Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the block by AcDbObjectId.

```
Acad::ErrorStatus SetBlockId(  
    const AcDbObjectId idBlock  
);
```

Parameters

Description

idBlock

Input block AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetLayerId Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the layer by AcDbObjectId.

```
Acad::ErrorStatus SetLayerId(  
    const AcDbObjectId idLayer  
);
```

Parameters

Description

idLayer

Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemBlock Class](#),

[AcMapDMThematicTableItemBlock Class](#)

AcMapDMThematicTableItemBlock:: SetScale Method

[AcMapDMThematicTableItemBlock Class](#) |

[AcMapDMThematicTableItemBlock Class](#)

Sets the scale.

```
virtual Acad::ErrorStatus SetScale(  
    double scale  
);
```

Parameters	Description
------------	-------------

scale	Input scale.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: ~AcMapDMThematicTableItemColor  
Destructor

[AcMapDMThematicTableItemColor Class](#) |  
[AcMapDMThematicTableItemColor Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemColor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor::

AcMapDMThematicTableItemColor Constructor

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemColor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor::

AcMapDMThematicTableItemColor Constructor

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemColor(  
    const AcMapDMThematicTableItemColor& other  
);
```

Parameters

Description

other

Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: Color Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Retrieves the color.

```
virtual AcCmColor Color() const;
```

Returns

Returns the color.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: dwgInFields Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: dwgOutFields Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: dxfInFields Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: dxfOutFields Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor:: Implementation Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemColor* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemColor Class](#), [AcMapDMThematicTableItemColor Class](#)

AcMapDMThematicTableItemColor::SetColor Method

[AcMapDMThematicTableItemColor Class](#) |

[AcMapDMThematicTableItemColor Class](#)

Sets the color.

```
virtual Acad::ErrorStatus SetColor(  
    const AcCmColor pColor  
);
```

Parameters	Description
------------	-------------

pColor	Input color.
--------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue::

~AcMapDMThematicTableItemDataValue Destructor

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMThematicTableItemDataValue\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue::

AcMapDMThematicTableItemDataValue Constructor

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemDataValue();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue::

AcMapDMThematicTableItemDataValue Constructor

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemDataValue(  
    const AcMapDMThematicTableItemDataValue& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: dwgInFields Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: dwgOutFields Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: dxfInFields Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: dxfOutFields Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: Implementation Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemDataValue* Implementation() const
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: SetValue Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Sets the value.

```
virtual Acad::ErrorStatus SetValue(  
    double value  
);
```

Parameters	Description
value	Input value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: Text Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Retrieves the text.

```
virtual const ACHAR* Text() const;
```

Returns

Returns the text if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemDataValue Class](#),

[AcMapDMThematicTableItemDataValue Class](#)

AcMapDMThematicTableItemDataValue:: Value Method

[AcMapDMThematicTableItemDataValue Class](#) |

[AcMapDMThematicTableItemDataValue Class](#)

Retrieves the value.

```
virtual double Value() const;
```

Returns

Returns the value if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch::~~AcMapDMThematicTableItemHatch  
Destructor

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemHatch();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch::

AcMapDMThematicTableItemHatch Constructor

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemHatch();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch::

AcMapDMThematicTableItemHatch Constructor

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemHatch(  
    const AcMapDMThematicTableItemHatch& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: Angle Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the angle.

```
virtual int Angle() const;
```

Returns

Returns the angle.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: Color Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the color.

```
virtual AcCmColor Color() const;
```

Returns

Returns the color.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: dwgInFields Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: dwgOutFields Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: dxfInFields Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: dxfOutFields Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: Implementation Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemHatch* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: LayerId Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the layer AcDbObjectId.

```
AcDbObjectId LayerId() const;
```

Returns

Returns the layer AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: LayerName Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the layer name.

```
virtual const ACHAR* LayerName() const;
```

Returns

Returns the layer name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: PatternName Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the pattern name.

```
virtual const ACHAR* PatternName() const;
```

Returns

Returns the pattern name if successful. or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: Scale Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Retrieves the scale.

```
virtual double Scale() const;
```

Returns

Returns the scale.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetAngle Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the angle.

```
virtual Acad::ErrorStatus SetAngle(  
    int angle  
);
```

Parameters	Description
------------	-------------

angle	Input angle.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetColor Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the color.

```
virtual Acad::ErrorStatus SetColor(  
    const AcCmColor color  
);
```

Parameters	Description
------------	-------------

color	Input color.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetLayerId Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the layer AcDbObjectId.

```
Acad::ErrorStatus SetLayerId(  
    AcDbObjectId layerId  
);
```

Parameters

Description

layerId

Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemHatch Class](#),

[AcMapDMThematicTableItemHatch Class](#)

AcMapDMThematicTableItemHatch:: SetScale Method

[AcMapDMThematicTableItemHatch Class](#) |

[AcMapDMThematicTableItemHatch Class](#)

Sets the scale.

```
virtual Acad::ErrorStatus SetScale(  
    double scale  
);
```

Parameters	Description
------------	-------------

scale	Input scale.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText::

~AcMapDMThematicTableItemLegendText Destructor

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Destroys an instance of this class.

**virtual** `~AcMapDMThematicTableItemLegendText()`;

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText::

AcMapDMThematicTableItemLegendText Constructor

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemLegendText();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText::

AcMapDMThematicTableItemLegendText Constructor

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemLegendText(  
    const AcMapDMThematicTableItemLegendText& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: Color Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the color of the cell.

```
virtual AcCmColor Color() const;
```

Returns

Returns the color.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: dwgInFields Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: dwgOutFields Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: dxfInFields Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: dxfOutFields Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: Height Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the height of the cell.

```
virtual double Height() const;
```

Returns

Returns the height.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: Implementation Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Returns the implementation object.

**virtual** AcMapDMImpThematicTableItemLegendText\* Implementation() **cons**

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: LayerId Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the layer AcDbObjectId of the cell.

```
AcDbObjectId LayerId() const;
```

Returns

Returns the layer AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: LayerName Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the layer name of the cell.

```
virtual const ACHAR* LayerName() const;
```

Returns

Returns the layer name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText::SetColor Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the color of the cell.

```
virtual Acad::ErrorStatus SetColor(  
    const AcCmColor color  
);
```

Parameters	Description
------------	-------------

color	Input color.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: SetHeight Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the height of the cell.

```
virtual Acad::ErrorStatus SetHeight(  
    double height  
);
```

Parameters	Description
height	Input height.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: SetLayerId Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the layer AcDbObjectId of the cell.

```
Acad::ErrorStatus SetLayerId(  
    const AcDbObjectId layerId  
);
```

Parameters

Description

layerId

Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: SetStyleId Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Sets the style AcDbObjectId of the cell.

```
Acad::ErrorStatus SetStyleId(  
    const AcDbObjectId styleId  
);
```

Parameters

Description

styleId

Input style AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: StyleId Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the style AcDbObjectId of the cell.

```
AcDbObjectId StyleId() const;
```

Returns

Returns the style AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: StyleName Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the style name of the cell.

```
virtual const ACHAR* StyleName() const;
```

Returns

Returns the style name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLegendText Class](#),

[AcMapDMThematicTableItemLegendText Class](#)

AcMapDMThematicTableItemLegendText:: Text Method

[AcMapDMThematicTableItemLegendText Class](#) |

[AcMapDMThematicTableItemLegendText Class](#)

Retrieves the text of the cell.

```
virtual const ACHAR* Text() const;
```

Returns

Returns the text if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle::

~AcMapDMThematicTableItemLinestyle Destructor

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMThematicTableItemLinestyle\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle::

AcMapDMThematicTableItemLinestyle Constructor

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemLinestyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle::

AcMapDMThematicTableItemLinestyle Constructor

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemLinestyle(  
    const AcMapDMThematicTableItemLinestyle& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: dwgInFields Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: dwgOutFields Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: dxfInFields Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: dxfOutFields Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: Implementation Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Returns the implementation object.

**virtual** AcMapDMImpThematicTableItemLinestyle\* Implementation() **const**

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: LayerId Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Retrieves the layer AcDbObjectId.

```
AcDbObjectId LayerId() const;
```

Returns

Returns the layer AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: LayerName Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Retrieves the layer name.

```
virtual const ACHAR* LayerName() const;
```

Returns

Returns the layer name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: LinetypeId Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Retrieves the linetype AcDbObjectId.

```
AcDbObjectId LinetypeId() const;
```

Returns

Returns the AcDbObjectId name if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineStyle Class](#),

[AcMapDMThematicTableItemLineStyle Class](#)

AcMapDMThematicTableItemLineStyle:: LinetypeName Method

[AcMapDMThematicTableItemLineStyle Class](#) |

[AcMapDMThematicTableItemLineStyle Class](#)

Retrieves the linetype name.

```
virtual const ACHAR* LinetypeName() const;
```

Returns

Returns the linetype name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: SetLayerId Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Sets the layer AcDbObjectId.

```
Acad::ErrorStatus SetLayerId(  
    const AcDbObjectId idLayer  
);
```

Parameters	Description
idLayer	Input layer name.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: SetLinetypeId Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Sets the linetype AcDbObjectId.

```
Acad::ErrorStatus SetLinetypeId(  
    const AcDbObjectId idLinetype  
);
```

Parameters

Description

idLinetype

Input linetype AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinestyle Class](#),

[AcMapDMThematicTableItemLinestyle Class](#)

AcMapDMThematicTableItemLinestyle:: SetWidth Method

[AcMapDMThematicTableItemLinestyle Class](#) |

[AcMapDMThematicTableItemLinestyle Class](#)

Sets the width.

```
virtual Acad::ErrorStatus SetWidth(  
    double width  
);
```

Parameters	Description
width	Input width.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineStyle Class](#),

[AcMapDMThematicTableItemLineStyle Class](#)

AcMapDMThematicTableItemLineStyle:: Width Method

[AcMapDMThematicTableItemLineStyle Class](#) |

[AcMapDMThematicTableItemLineStyle Class](#)

Retrieves the width.

```
virtual double width() const;
```

Returns

Returns the width.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype::

~AcMapDMThematicTableItemLinetype Destructor

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemLinetype();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype::

AcMapDMThematicTableItemLinetype Constructor

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemLinetype();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype::

AcMapDMThematicTableItemLinetype Constructor

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemLinetype(  
    const AcMapDMThematicTableItemLinetype& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: dwgInFields Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: dwgOutFields Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: dxfInFields Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: dxfOutFields Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: Implementation Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemLinetype* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: Linetype Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Retrieves the linetype.

```
virtual const ACHAR * Linetype() const;
```

Returns

Returns the linetype name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: LinetypeId Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Retrieves the linetype AcDbObjectId.

```
AcDbObjectId LinetypeId() const;
```

Returns

Returns the linetype AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLinetype Class](#),

[AcMapDMThematicTableItemLinetype Class](#)

AcMapDMThematicTableItemLinetype:: SetLinetypeId Method

[AcMapDMThematicTableItemLinetype Class](#) |

[AcMapDMThematicTableItemLinetype Class](#)

Sets the linetype AcDbObjectId.

```
Acad::ErrorStatus SetLinetypeId(  
    const AcDbObjectId idLinetype  
);
```

Parameters

Description

idLinetype

Input linetype AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight::

~AcMapDMThematicTableItemLineweight Destructor

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemLineweight();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight::

AcMapDMThematicTableItemLineweight Constructor

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemLineweight();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight::

AcMapDMThematicTableItemLineweight Constructor

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemLineweight(  
    const AcMapDMThematicTableItemLineweight& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: dwgInFields Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: dwgOutFields Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: dxfInFields Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: dxfOutFields Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: Implementation Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Returns the implementation object.

**virtual** AcMapDMImpThematicTableItemLineweight\* Implementation() **cons**

Returns

Returns the implementation object, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: Lineweight Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Retrieves the lineweight.

```
virtual AcDb::Lineweight Lineweight() const;
```

Returns

Returns the lineweight.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemLineweight Class](#),

[AcMapDMThematicTableItemLineweight Class](#)

AcMapDMThematicTableItemLineweight:: SetLineweight Method

[AcMapDMThematicTableItemLineweight Class](#) |

[AcMapDMThematicTableItemLineweight Class](#)

Sets the lineweight.

```
virtual Acad::ErrorStatus SetLineweight(  
    AcDb::Lineweight lineweight  
);
```

Parameters	Description
lineweight	Input lineweight.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle::

~AcMapDMThematicTableItemPlotstyle Destructor

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMThematicTableItemPlotstyle\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle::

AcMapDMThematicTableItemPlotstyle Constructor

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemPlotstyle();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle::

AcMapDMThematicTableItemPlotstyle Constructor

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemPlotstyle(  
    const AcMapDMThematicTableItemPlotstyle& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: dwgInFields Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: dwgOutFields Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: dxfInFields Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: dxfOutFields Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle:: Implementation Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemPlotstyle* Implementation() const
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemPlotstyle Class](#),

[AcMapDMThematicTableItemPlotstyle Class](#)

AcMapDMThematicTableItemPlotstyle::PlotstyleName Method

[AcMapDMThematicTableItemPlotstyle Class](#) |

[AcMapDMThematicTableItemPlotstyle Class](#)

Retrieves the plotstyle name.

```
virtual const ACHAR* PlotstyleName() const;
```

Returns

Returns the plotstyle name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: ~AcMapDMThematicTableItemText  
Destructor

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableItemText();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText::

AcMapDMThematicTableItemText Constructor

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableItemText();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText::

AcMapDMThematicTableItemText Constructor

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Constructs an instance of this class by using another instance of this class.

```
AcMapDMThematicTableItemText(  
    const AcMapDMThematicTableItemText& other  
);
```

Parameters	Description
other	Input other instance of this class.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Angle Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the angle.

```
virtual int Angle() const;
```

Returns

Returns the angle.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Color Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the color.

```
virtual AcCmColor Color() const;
```

Returns

Returns the color.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: dwgInFields Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: dwgOutFields Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: dxfInFields Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: dxfOutFields Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Height Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the height.

```
virtual double Height() const;
```

Returns

Returns the height.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Implementation Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Returns the implementation object.

```
virtual AcMapDMImpThematicTableItemText* Implementation() const;
```

Returns

Returns the implementation object, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: InsertPoint Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the insertion point.

**virtual** [AcMapThematicTableItemInsertionPoint](#) InsertPoint() **const**;

Returns

Returns an [AcMapThematicTableItemInsertionPoint](#) value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Justification Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the justification.

**virtual** [AcMapThematicTableItemJustificationType](#) Justification() **const**  
Returns

Returns an [AcMapThematicTableItemJustificationType](#) value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: LayerId Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the layer AcDbObjectId.

```
AcDbObjectId LayerId() const;
```

Returns

Returns the layer AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: LayerName Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the layer name.

```
virtual const ACHAR* LayerName() const;
```

Returns

Returns the layer name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetAngle Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the angle.

```
virtual Acad::ErrorStatus SetAngle(  
    int angle  
);
```

Parameters	Description
angle	Input angle.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetColor Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the color.

```
virtual Acad::ErrorStatus SetColor(  
    const AcCmColor color  
);
```

Parameters	Description
------------	-------------

color	Input color.
-------	--------------

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetHeight Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the height.

```
virtual Acad::ErrorStatus SetHeight(  
    double height  
);
```

Parameters	Description
height	Input height.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetInsertPoint Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the insertion point.

```
virtual Acad::ErrorStatus SetInsertPoint(  
    AcMapThematicTableItemInsertionPoint insertPoint  
);
```

Parameters

Description

insertPoint

Input [AcMapThematicTableItemInsertionPoint](#)value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetJustification Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the justification type.

```
virtual Acad::ErrorStatus SetJustification(  
    AcMapThematicTableItemJustificationType justification  
);
```

Parameters	Description
justification	Input <a href="#">AcMapThematicTableItemJustificationType</a> value.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetLayerId Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the layer AcDbObjectId.

```
Acad::ErrorStatus SetLayerId(  
    const AcDbObjectId idLayer  
);
```

Parameters	Description
idLayer	Input layer AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: SetStyleId Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Sets the style AcDbObjectId.

```
Acad::ErrorStatus SetStyleId(  
    const AcDbObjectId idStyle  
);
```

Parameters	Description
idStyle	Input style AcDbObjectId.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: StyleId Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the style AcDbObjectId.

```
AcDbObjectId StyleId() const;
```

Returns

Returns the style AcDbObjectId if successful, or AcDbObjectId::kNull otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: StyleName Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the style name.

```
virtual const ACHAR* StyleName() const;
```

Returns

Returns the style name if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableItemText Class](#), [AcMapDMThematicTableItemText Class](#)

AcMapDMThematicTableItemText:: Text Method

[AcMapDMThematicTableItemText Class](#) | [AcMapDMThematicTableItemText Class](#)

Retrieves the text.

```
virtual const ACHAR* Text() const;
```

Returns

Returns the text if successful, or NULL otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpAnnotation Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Annotation value type.

[AcMapDMThematicTableItemAnnotation](#) \* mpAnnotation;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpBlock Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Block value type.

[AcMapDMThematicTableItemBlock](#) \* mpBlock;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpColor Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Color value type.

[AcMapDMThematicTableItemColor](#) \* mpColor;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpDataValue Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Text/value-pair value type.

[AcMapDMThematicTableItemDataValue](#) \* mpDataValue;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

[AcMapDMThematicTableRow:: mpHatch Data Member](#)

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Hatch value type.

[AcMapDMThematicTableItemHatch](#) \* mpHatch;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpLegendText Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Legend-text value type.

[AcMapDMThematicTableItemLegendText](#) \* mpLegendText;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpLinestyle Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Linestyle value type.

[AcMapDMThematicTableItemLinestyle](#) \* mpLinestyle;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpLinetype Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Linetype value type.

[AcMapDMThematicTableItemLinetype](#) \* mpLinetype;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpLineweight Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Lineweight value type.

[AcMapDMThematicTableItemLineweight](#) \* mpLineweight;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpPlotstyle Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Plotstyle value type.

[AcMapDMThematicTableItemPlotstyle](#) \* mpPlotstyle;

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

AcMapDMThematicTableRow:: mpText Data Member

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Text value type.

```
AcMapDMThematicTableItemText * mpText;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)  
AcMapDMThematicTableRow:: ~AcMapDMThematicTableRow Destructor  
[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMThematicTableRow();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)  
AcMapDMThematicTableRow:: AcMapDMThematicTableRow Constructor  
[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Constructs an instance of this class.

```
AcMapDMThematicTableRow();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

[AcMapDMThematicTableRow:: closeRow Method](#)

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Closes all members.

```
Acad::ErrorStatus closeRow();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMThematicTableRow Class](#), [AcMapDMThematicTableRow Class](#)

[AcMapDMThematicTableRow:: Init Method](#)

[AcMapDMThematicTableRow Class](#) | [AcMapDMThematicTableRow Class](#)

Initializes the member variables.

```
void Init();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)  
AcMapDMTopoElement:: ~AcMapDMTopoElement Destructor  
[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMTopoElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)  
AcMapDMTopoElement:: AcMapDMTopoElement Constructor  
[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Constructs an instance of this class.

```
AcMapDMTopoElement ( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: AcquireEntities Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Runs the topological query against the current drawing and acquires both selected entities and topology objects.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: ClonesObjectsFromExternalSource Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Always returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement::DismissEntities Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Erases queried objects.

```
virtual Acad::ErrorStatus DismissEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: dwgInFields Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: dwgOutFields Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: dxfInFields Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: dxfOutFields Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: GetAcquisitionCriteria Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters

Description

pDataSourceDsc

Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)

AcMapDMTopoElement:: OnMapProjectInitialized Method

[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoElement Class](#), [AcMapDMTopoElement Class](#)  
AcMapDMTopoElement:: SetAcquisitionCriteria Method  
[AcMapDMTopoElement Class](#) | [AcMapDMTopoElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters	Description
pDataSourceDsc	Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor::

~AcMapDMTopoElementDataSourceDescriptor Destructor

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Destroys an instance of this class.

```
~AcMapDMTopoElementDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoElementDataSourceDescriptor Class](#),

[AcMapDMTopoElementDataSourceDescriptor Class](#)

AcMapDMTopoElementDataSourceDescriptor::

AcMapDMTopoElementDataSourceDescriptor Constructor

[AcMapDMTopoElementDataSourceDescriptor Class](#) |

[AcMapDMTopoElementDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMTopoElementDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor::

~AcMapDMTopoQueryDataSourceDescriptor Destructor

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Destroys an instance of this class.

**virtual** [~AcMapDMTopoQueryDataSourceDescriptor\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor::

AcMapDMTopoQueryDataSourceDescriptor Constructor

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Constructs an instance of this class.

```
AcMapDMTopoQueryDataSourceDescriptor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: GetQuery Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Retrieves the query definition.

```
virtual Acad::ErrorStatus GetQuery(  
    struct resbuf*& pRb  
) const;
```

Parameters

Description

pRb

Output resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryDataSourceDescriptor Class](#),

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

AcMapDMTopoQueryDataSourceDescriptor:: SetQuery Method

[AcMapDMTopoQueryDataSourceDescriptor Class](#) |

[AcMapDMTopoQueryDataSourceDescriptor Class](#)

Sets the query definition.

```
virtual Acad::ErrorStatus SetQuery(  
    const struct resbuf* pRb  
);
```

Parameters

Description

pRb

Input resbuf object that contains the query.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: ~AcMapDMTopoQueryElement Destructor  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapDMTopoQueryElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: AcMapDMTopoQueryElement Constructor  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Constructs an instance of this class.

```
AcMapDMTopoQueryElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)

[AcMapDMTopoQueryElement:: AcquireEntities Method](#)

[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Runs the topological query and acquires both selected entities and topology objects.

```
virtual Acad::ErrorStatus AcquireEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: ClonesObjectsFromExternalSource Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Clones objects from the source drawings.

```
virtual bool ClonesObjectsFromExternalSource() const;
```

Returns

Always returns true.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
[AcMapDMTopoQueryElement::DismissEntities Method](#)  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Erases entities that are part of this query element.

```
virtual Acad::ErrorStatus DismissEntities();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

See also [AcquireEntities\(\)](#).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: DismissStylization Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Dismisses the current stylization.

```
virtual Acad::ErrorStatus DismissStylization(  
    bool bDismissEntities  
);
```

Parameters	Description
bDismissEntities	Input true to unstylize the entities.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: dwgInFields Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: dwgOutFields Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: dxfInFields Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: dxfOutFields Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overloaded function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write in the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)

[AcMapDMTopoQueryElement:: EmptyCookieJar Method](#)

[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Called during database destruction to trigger the deletion of topology-managed cookies.

```
void EmptyCookieJar();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: EnableStyle Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Enables or disables a style reference.

```
virtual Acad::ErrorStatus EnableStyle(  
    AcDbObjectId styleRefId,  
    bool bNewEnableStatus  
);
```

Parameters	Description
styleRefId	Input style reference ID.
bNewEnableStatus	Input new enable status.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: GetAcquisitionCriteria Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Retrieves the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus GetAcquisitionCriteria(  
    AcMapDMDataSourceDescriptor* pDataSourceDsc  
) const;
```

Parameters	Description
pDataSourceDsc	Output pointer to the data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: OnMapProjectInitialized Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Invoked when an AutoCAD Map project is initialized.

```
virtual bool OnMapProjectInitialized(  
    AcMapProject* pMapProject  
);
```

Parameters	Description
pMapProject	Input AutoCAD Map project.

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: RemoveStyle Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Removes a style reference from this element.

```
virtual Acad::ErrorStatus RemoveStyle(  
    AcDbObjectId styleRefId  
);
```

Parameters	Description
styleRefId	Input style reference ID.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: SetAcquisitionCriteria Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Sets the query definition's data-source descriptor.

```
virtual Acad::ErrorStatus SetAcquisitionCriteria(  
    const AcMapDMDataSourceDescriptor* pDataSourceDsc  
);
```

Parameters	Description
pDataSourceDsc	Input pointer to a data-source descriptor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This function determines which entities become part of this element. Current-drawing topologies are ignored.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: SetVisible Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Makes the entities that are part of this element visible or invisible.

```
virtual Acad::ErrorStatus SetVisible(  
    bool bNewVal,  
    double dScale  
);
```

Parameters	Description
bNewVal	Input true to make the entities visible, or false to make them invisible.
dScale	Input scale at which to make entities visible.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapDMTopoQueryElement Class](#), [AcMapDMTopoQueryElement Class](#)  
AcMapDMTopoQueryElement:: UpdateStylization Method  
[AcMapDMTopoQueryElement Class](#) | [AcMapDMTopoQueryElement Class](#)

Stylizes the current or updated selection at the current scale.

```
virtual Acad::ErrorStatus UpdateStylization(  
    bool bAcquireEntities  
);
```

### Parameters

### Description

bAcquireEntities

Input true to acquire entities as part of update operation.

### Returns

Returns Acad::eOk if successful. Returns Acad::eAmbiguousOutput if bAcquireEntities is true but the previous selection has not been dismissed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CExportResults Structure](#), [CExportResults Structure](#), [AcMapIE Namespace](#)  
CExportResults:: m\_ulEntitiesExported Data Member  
[CExportResults Structure](#) | [CExportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesExported, a member of class CExportResults.

```
unsigned long m_ulEntitiesExported;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CExportResults Structure](#), [CExportResults Structure](#), [AcMapIE Namespace](#)  
CExportResults:: m\_ulEntitiesSkippedCouldntTransform Data Member  
[CExportResults Structure](#) | [CExportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesSkippedCouldntTransform, a member of class CExportResults.

```
unsigned long m_ulEntitiesSkippedCouldntTransform;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CExportResults Structure](#), [CExportResults Structure](#), [AcMapIE Namespace](#)

[CExportResults:: CExportResults Constructor](#)

[CExportResults Structure](#) | [CExportResults Structure](#) | [AcMapIE Namespace](#)

This is CExportResults, a member of class CExportResults.

```
CExportResults();
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

CImportResults:: m\_ulEntitiesImported Data Member

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesImported, a member of class CImportResults.

```
unsigned long m_ulEntitiesImported;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)  
CImportResults:: m\_ulEntitiesSkippedCouldntTransform Data Member  
[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesSkippedCouldntTransform, a member of class CImportResults.

```
unsigned long m_ulEntitiesSkippedCouldntTransform;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

CImportResults:: m\_bEvaluateHatchFailed Data Member

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_bEvaluateHatchFailed, a member of class CImportResults.

```
bool m_bEvaluateHatchFailed;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)  
CImportResults:: m\_ulEntitiesWithColorCloseToBackground Data Member  
[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesWithColorCloseToBackground, a member of class CImportResults.

```
unsigned long m_ulEntitiesWithColorCloseToBackground;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

CImportResults:: m\_ulEntitiesClassified Data Member

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesClassified, a member of class CImportResults.

```
unsigned long m_ulEntitiesClassified;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

[CImportResults:: m\\_ulEntitiesNotClassifiable Data Member](#)

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulEntitiesNotClassifiable, a member of class CImportResults.

```
unsigned long m_ulEntitiesNotClassifiable;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

CImportResults:: m\_ulOutOfRangeNotFixed Data Member

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is m\_ulOutOfRangeNotFixed, a member of class CImportResults.

```
unsigned long m_ulOutOfRangeNotFixed;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[CImportResults Structure](#), [CImportResults Structure](#), [AcMapIE Namespace](#)

[CImportResults:: CImportResults Constructor](#)

[CImportResults Structure](#) | [CImportResults Structure](#) | [AcMapIE Namespace](#)

This is CImportResults, a member of class CImportResults.

```
CImportResults();
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

[AcMapIEColumn:: ~AcMapIEColumn Destructor](#)

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEColumn();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

[AcMapIEColumn:: ColumnClassMapping Method](#)

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Returns the name of the feature class property that is mapped to this column.

```
virtual const ACHAR* ColumnClassMapping() const = 0;
```

Returns

Returns the feature class property string. The format of this string is *category:property*, where both the *category* and *property* substrings are extended symbol names that can contain special characters. Furthermore, *category* can take the form *OD:table\_name*, where *table\_name* is the name of an object data table. If you are developing international application, note that the substring "OD:" is localizable.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

[AcMapIEColumn::ColumnDataMapping Method](#)

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Returns the name of the column in the AutoCAD Map table that this incoming column is mapped to.

```
virtual const ACHAR* ColumnDataMapping(  
    AcMapIE::ImportDataMapping& colMappingType  
) const = 0;
```

Parameters	Description
colMappingType	Output ImportDataMappingtype of column mapping. This value is provided as a convenience; the type was set previously in the input layer in the AcMapIEInputLayerclass.

## Returns

Returns the name of the mapped column.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEColumn Class](#), [AcMapIEColumn Class](#)

AcMapIEColumn::ColumnName Method

[AcMapIEColumn Class](#) | [AcMapIEColumn Class](#)

Returns the (fixed) name of this incoming column.

```
virtual const ACHAR* ColumnName() const = 0;
```

Returns

Returns the name of this column.

## Remarks

The allowable characters are: a-z A-Z 0-9 . { } \_ -.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)  
AcMapIEColumnIterator:: ~AcMapIEColumnIterator Destructor  
[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEColumnIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)

[AcMapIEColumnIterator:: Done Method](#)

[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)

[AcMapIEColumnIterator:: Get Method](#)

[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Retrieves the current element in the iteration.

```
virtual bool Get(  
    AcMapIEColumn*& pColumn  
) = 0;
```

Parameters

Description

pColumn

Output pointer to the current AcMapIEColumn.

Returns

Returns true if the current element is valid, or false if the iteration has no more elements to retrieve.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)

[AcMapIEColumnIterator::Rewind Method](#)

[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEColumnIterator Class](#), [AcMapIEColumnIterator Class](#)

[AcMapIEColumnIterator:: Step Method](#)

[AcMapIEColumnIterator Class](#) | [AcMapIEColumnIterator Class](#)

Advances to the next element in the iteration.

```
virtual void Step() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: AddReactor Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Adds a reactor to this exporter. You must derive a reactor from the virtual base class [AcMapIEExportReactor](#). Reactors are triggered through only the API; an end-user cannot trigger a reactor by using the AutoCAD Map user interface. Reactors are live until they are removed explicitly with RemoveReactor() or until AutoCAD Map exits. An exporter is a singleton; all callers add reactors to the same exporter instance. If an exporter has multiple reactors, they are called in the order added.

```
virtual AcMapIE::ErrCode AddReactor(  
    AcMapIEExportReactor * pReactor  
) = 0;
```

Parameters

Description

pReactor

Input [AcMapIEExportReactor](#) export reactor to add.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::ClosedPolylinesAsPolygons Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Determines whether closed polylines are exported as polygons.

```
virtual bool ClosedPolylinesAsPolygons() const = 0;
```

Returns

Returns true if closed polylines are exported as polygons, or false if they are exported as polylines.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter::CountObjects Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Counts the number of objects to export in the current drawing.

```
virtual AcMapIE::ErrCode CountObjects(  
    int& numPoints,  
    int& numLines,  
    int& numPolygons,  
    int& numText,  
    int& numTotal  
) = 0;
```

Parameters	Description
numPoints	Output number of points to export.
numLines	Output number of lines to export.
numPolygons	Output number of polygons to export.
numText	Output number of text entities to export.
numTotal	Output total number of entities to export.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_OutOfMemory if system resources are insufficient. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

The output values depend on the Layer filter, Feature class filter, and selection set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter::DiscretizationAngle Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the discretization angle. The discretization angle setting is stored in the file *MapExport.ini*, and labeled SegmentationDegrees.

```
virtual bool DiscretizationAngle(  
    double& dDegrees  
) const = 0;
```

Parameters

Description

dDegrees

Output discretization angle, in decimal degrees.

Returns

Returns true if the exporter was initialized; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::DriverOptions Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the current driver options. Driver options can be set in the file *MapExport.ini*, by calling [InvokeDriverOptionsDialog\(\)](#), or by loading a profile. Use this function to first retrieve the current driver options before adding driver options with [SetDriverOptions\(\)](#).

```
virtual AcMapIE::ErrCode DriverOptions(  
    AcMapIENameValueIterator*& nameValueIter  
) = 0;
```

Parameters	Description
nameValueIter	Output driver options, as <a href="#">AcMapIENameValueIterator</a> name-value pairs. The caller is responsible for deleting this object.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::InvokeDriverOptionsDialog Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Invokes the driver-specific Driver Options dialog box. Not all drivers have dialog boxes, so this function may finish immediately, with the returned error code indicating the status of the dialog box. Any setting changes that an AutoCAD Map user makes in this dialog box are reflected in subsequent calls to `DriverOptions()`. You can set driver options programmatically by using `SetDriverOptions()`. Any driver options in the file *MapExport.ini* are read and available in the set of driver options returned by `DriverOptions()`.

```
virtual AcMapIE::ErrCode InvokeDriverOptionsDialog() = 0;
```

Returns

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_DialogNotAvailable` if no driver options dialog box exists for this format. Returns `AcMapIE::ErrCode kErr_Fail` if the dialog box cannot be created or if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: Export Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Performs the export.

```
virtual AcMapIE::ErrCode Export(  
    AcMapIE::CExportResults& results,  
    bool bUseProgressDialog = false  
) = 0;
```

Parameters	Description
results	Output CExportResultsexport results.
bUseProgressDialog	Input true to display AutoCAD Map's progress dialog box during the export, or false to suppress this dialog box.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_OutOfMemory if system resources are insufficient. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

This operation is time-consuming for a large number of entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter:: ExportAll Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Determines whether a selection set is ignored during export.

```
virtual bool ExportAll() const = 0;
```

Returns

Returns false if a selection set is used; returns true if a selection set and all filtering is to be ignored.

Remarks

The default is to not export all but to use a selection set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::ExportDataMappings Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the source-column and output-column data mappings for export. The output collection will be empty if you have not first loaded an export profile file with LoadEPF().

```
virtual AcMapIE::ErrCode ExportDataMappings(  
    AcMapIEExpressionTargetIterator*& expTargetIter  
) = 0;
```

Parameters	Description
expTargetIter	Output data mappings, as <a href="#">AcMapIEExpressionTargetIterator</a> expression-target pairs. The caller is responsible for deleting this object.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::FeatureClassFilter Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the feature classes as a list of comma-separated patterns.

```
virtual const ACHAR* FeatureClassFilter() const = 0;
```

Returns

Returns a list of feature classes, as a string.

## Remarks

The patterns follow the rules for `acutWcMatch`. The comma character is not permitted in extended symbol names, so the list is unambiguous.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::FormatName Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Returns the name of the format set during initialization. The format name is set with Init().

```
virtual const ACHAR* FormatName() const = 0;
```

Returns

Returns the format name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: LayerFilter Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the layers as a list of comma-separated patterns.

```
virtual const ACHAR* LayerFilter() const = 0;
```

Returns

Returns a list of layers, as a string.

## Remarks

The patterns follow the rules for `acutWcMatch`. The comma character is not permitted in extended symbol names, so the list is unambiguous.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::LayerLevelMapping Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the settings for mapping layers in the drawing to DGN levels.

```
virtual AcMapIE::ErrCode LayerLevelMapping(  
    bool& bUse,  
    AcMapIENameValueIterator*& mapping  
) = 0;
```

Parameters	Description
bUse	Output true if layer mapping is enabled; otherwise, false.
mapping	Output mapping, as <a href="#">AcMapIENameValueIterator</a> name-value pairs. In each name-value pair, an AutoCAD Map layer name is the name component and its corresponding level number is the value component. The caller is responsible for deleting this object.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

## Remarks

DGN supports 64 levels, numbered 0 to 63. Level 0 is reserved, so the default mapping starts with level 1.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: RemoveReactor Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Removes a reactor from this exporter.

```
virtual AcMapIE::ErrCode RemoveReactor(  
    AcMapIEExportReactor * pReactor  
) = 0;
```

Parameters

Description

pReactor

Input [AcMapIEExportReactor](#) export reactor to remove.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if the reactor does not already exist. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SelectionSet Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the selection set.

```
virtual AcMapIE::ErrCode SelectionSet(  
    ads_name& serset  
) const = 0;
```

Parameters	Description
seset	Output name of the ADS (AutoCAD Development System) selection set. Do not free this object.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetClosedPolylinesAsPolygons Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Changes the setting for exporting closed polylines as polygons.

```
virtual AcMapIE::ErrCode SetClosedPolylinesAsPolygons(  
    bool bPolygon  
) = 0;
```

Parameters

Description

bPolygon

Input true to export closed polylines as polygons;  
otherwise, false to export them as polylines.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetDiscretizationAngle Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the discretization angle. The discretization angle setting is stored in the file *MapExport.ini*, and labeled SegmentationDegrees.

```
virtual bool SetDiscretizationAngle(  
    double dDegrees  
) = 0;
```

Parameters

Description

dDegrees

Input discretization angle to set, expressed in decimal degrees.

Returns

Returns true if the exporter was initialized; otherwise returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetDriverOptions Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the driver options. Typically, you should retrieve the current driver options from DriverOptions() and make changes as needed by using this function.

```
virtual AcMapIE::ErrCode SetDriverOptions(  
    AcMapIENameValueIterator*& nameValueIter  
) = 0;
```

Parameters	Description
nameValueIter	Input driver options, as <a href="#">AcMapIENameValueIterator</a> name-value pairs.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_OutOfMemory if system resources are insufficient. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

[AcMapIEExporter:: SetExportAll Method](#)

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Changes the setting for using a selection set during export.

```
virtual AcMapIE::ErrCode SetExportAll(  
    bool bTrue  
) = 0;
```

Parameters

Description

bTrue

Input true to export all entities in a drawing ignoring the selection set and filters; otherwise, false.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Remarks

A default selection set (containing the entire drawing) is used if no selection set is set explicitly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::SetExportDataMappings Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the source-column and output-column data mappings for export.

```
virtual AcMapIE::ErrCode SetExportDataMappings(  
    AcMapIEExpressionTargetIterator*& expTargetIter  
) = 0;
```

Parameters	Description
expTargetIter	Input export data mappings, as <a href="#">AcMapIEExpressionTargetIterator</a> expression-target pairs.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_ConflictWithKey if a target column has the same name as the unique key column; see UseUniqueKeyField(). Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

The valid characters for column names are: a-z A-Z 0-9 . { } \_ -. (Including accented and multibyte characters.) The dot and brace characters have special meaning and should be avoided except in special circumstances.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetLayerLevelMapping Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Changes the settings for mapping layers in the drawing to DGN levels.

```
virtual AcMapIE::ErrCode SetLayerLevelMapping(  
    bool bUse,  
    const AcMapIENAMEValueIterator*& mapping  
) = 0;
```

Parameters	Description
bUse	Input true to enable layer mapping; otherwise, false.
mapping	Input mapping to set, as <a href="#">AcMapIENAMEValueIterator</a> name-value pairs. In each name-value pair, specify an AutoCAD Map layer name in the name component, and its corresponding level number as a string in the value component.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter:: SetSelectionSet Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Sets the ADS (AutoCAD Development System) selection set.

```
virtual AcMapIE::ErrCode SetSelectionSet(  
    ads_name& other  
) = 0;
```

Parameters	Description
other	Input selection set.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Remarks

The exporter is responsible for freeing the selection set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExporter Class](#), [AcMapIEExporter Class](#)

AcMapIEExporter::TargetCoordSys Method

[AcMapIEExporter Class](#) | [AcMapIEExporter Class](#)

Retrieves the target coordinate system for the exported data.

```
virtual const ACHAR* TargetCoordSys() const = 0;
```

Returns

Returns the name of the target coordinate system, as the system's Mentor abbreviation.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExportReactor Class](#), [AcMapIEExportReactor Class](#)  
AcMapIEExportReactor::~~AcMapIEExportReactor Destructor  
[AcMapIEExportReactor Class](#) | [AcMapIEExportReactor Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEExportReactor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExportReactor Class](#), [AcMapIEExportReactor Class](#)  
AcMapIEExportReactor:: AcMapIEExportReactor Constructor  
[AcMapIEExportReactor Class](#) | [AcMapIEExportReactor Class](#)

Constructs an instance of this class.

```
AcMapIEExportReactor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExportReactor Class](#), [AcMapIEExportReactor Class](#)

AcMapIEExportReactor::RecordExported Method

[AcMapIEExportReactor Class](#) | [AcMapIEExportReactor Class](#)

Invoked after an entity is exported.

```
virtual void RecordExported(  
    AcDbEntity* pEnt  
);
```

Parameters	Description
pEnt	Input entity.
Returns	

Returns nothing.

Remarks

You can use this function to track exported entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEExportReactor Class](#), [AcMapIEExportReactor Class](#)

[AcMapIEExportReactor::RecordReadyForExport Method](#)

[AcMapIEExportReactor Class](#) | [AcMapIEExportReactor Class](#)

Invoked before an entity is exported.

```
virtual bool RecordReadyForExport(  
    AcDbEntity* pEnt  
);
```

Parameters	Description
pEnt	Input entity.

Returns

Returns true if the specified entity should be exported; otherwise, returns false.

## Remarks

This entity is not a const, so you can modify it as needed. The return value indicates whether an entity is ready to export. If this function returns false, the entity is not exported and no further reactors in the chain are called. If this function returns true for all reactors in the chain, the entity is exported.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator::~~AcMapIEExpressionTargetIterator  
Destructor

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEExpressionTargetIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Clear Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Removes all elements from the collection.

```
virtual void Clear() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Done Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator::Rewind Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEExpressionTargetIterator Class](#), [AcMapIEExpressionTargetIterator Class](#)

AcMapIEExpressionTargetIterator:: Step Method

[AcMapIEExpressionTargetIterator Class](#) | [AcMapIEExpressionTargetIterator Class](#)

Advances to the next element in the iteration.

```
virtual void Step() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)  
AcMapIEFormat:: ~AcMapIEFormat Destructor  
[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEFormat();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)

[AcMapIEFormat::Description Method](#)

[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Retrieves a brief description of this format.

```
virtual const ACHAR* Description() const = 0;
```

Returns

Returns the description of this format.

## Remarks

This description is suitable for the Files Of Type drop-down box in AutoCAD Map's Import or Export dialog box.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)

[AcMapIEFormat:: Extension Method](#)

[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Retrieves the filename extension that AutoCAD Map expects to be used for this format.

```
virtual const ACHAR* Extension() const = 0;
```

Returns

Returns the filename extension used for this format, as a wildcard file specification such as "\*CATD.DDF" or "\*.dgn".

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)

AcMapIEFormat::FormatName Method

[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Returns the unique name which AutoCAD Map uses to refer to this format. This format name is not translated and is used in the AutoCAD Map configuration files *MapImport.ini*, *MapExport.ini*, and *MapForeignFileProperties.ini*.

```
virtual const ACHAR* FormatName() const = 0;
```

Returns

Returns the unique name of this format.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)

AcMapIEFormat:: HasDriverOptionsDialog Method

[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Determines whether a Driver Options dialog box exists for this format.

```
virtual bool HasDriverOptionsDialog() const = 0;
```

Returns

Returns true if a Driver Options dialog box exists for this format; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormat Class](#), [AcMapIEFormat Class](#)

AcMapIEFormat::StorageType Method

[AcMapIEFormat Class](#) | [AcMapIEFormat Class](#)

Returns the storage type of this format.

```
virtual AcMapIE::StorageType StorageType() const = 0;
```

Returns

Returns the StorageType storage type of this format. For export formats, this value can be any format except kStorage\_FileMultiSelect. For import formats, this value can be kStorage\_FileAllEntityTypes, kStorage\_FileMultiSelect, kStorage\_FolderNoPrefix, or kStorage\_Database.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormatIterator Class](#), [AcMapIEFormatIterator Class](#)

[AcMapIEFormatIterator:: Done Method](#)

[AcMapIEFormatIterator Class](#) | [AcMapIEFormatIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormatIterator Class](#), [AcMapIEFormatIterator Class](#)

[AcMapIEFormatIterator:: Get Method](#)

[AcMapIEFormatIterator Class](#) | [AcMapIEFormatIterator Class](#)

Retrieves the current element in the iteration.

```
virtual bool Get(  
    AcMapIEFormat*& pFormat  
) = 0;
```

Parameters

Description

pFormat

Output pointer to the current AcMapIEFormat. The caller is responsible for deleting this object.

Returns

Returns true if the current element is valid, or false if the iteration has no more elements to retrieve.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormatIterator Class](#), [AcMapIEFormatIterator Class](#)

[AcMapIEFormatIterator::Rewind Method](#)

[AcMapIEFormatIterator Class](#) | [AcMapIEFormatIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEFormatIterator Class](#), [AcMapIEFormatIterator Class](#)

[AcMapIEFormatIterator:: Step Method](#)

[AcMapIEFormatIterator Class](#) | [AcMapIEFormatIterator Class](#)

Advances to the next element in the iteration.

```
virtual void Step() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter::AddReactor Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Adds a reactor to this importer. You must derive a reactor from the virtual base class [AcMapIEImportReactor](#). Reactors are triggered through only the API; an end-user cannot trigger a reactor by using the AutoCAD Map user interface. Reactors are live until they are removed explicitly with `RemoveReactor()` or until AutoCAD Map exits. An importer is a singleton; all callers add reactors to the same importer instance. If an importer has multiple reactors, they are called in the order added.

```
virtual AcMapIE::ErrCode AddReactor(  
    AcMapIEImportReactor * pReactor  
) = 0;
```

Parameters	Description
pReactor	Input <a href="#">AcMapIEImportReactor</a> import reactor to add.
Returns	

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_Fail` if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: AuditClassifiedAfterImport Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Determines whether classified properties are audited after import.

```
virtual bool AuditClassifiedAfterImport() const = 0;
```

Returns

Returns true if classified-property auditing is turned on, or false if it is turned off.

Remarks

An audit determines whether each classified-property value is out of range and, if so, sets it to its default value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: DriverOptions Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Retrieves the current driver options. Driver options can be set in the file *MapImport.ini*, by calling [InvokeDriverOptionsDialog\(\)](#), or by loading a profile. Use this function to first retrieve the current driver options before adding driver options with [SetDriverOptions\(\)](#).

```
virtual AcMapIE::ErrCode DriverOptions(  
    AcMapIENameValueIterator*& pDriverOptions  
) = 0;
```

Parameters	Description
pDriverOptions	Output driver options, as <a href="#">AcMapIENameValueIterator</a> name-value pairs. The caller is responsible for deleting this object.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if the process failed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter::InvokeDriverOptionsDialog Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Invokes the driver-specific Driver Options dialog box. Not all drivers have dialog boxes, so this function may finish immediately, with the returned error code indicating the status of the dialog box. Any setting changes that an AutoCAD Map user makes in this dialog box are reflected in subsequent calls to `DriverOptions()`. You can set driver options programmatically by using `SetDriverOptions()`. Any driver options in the file *MapImport.ini* are read and available in the set of driver options returned by `DriverOptions()`.

```
virtual AcMapIE::ErrCode InvokeDriverOptionsDialog(  
    bool& bSchemaHasChanged  
) = 0;
```

Parameters	Description
<code>bSchemaHasChanged</code>	Output true if the schema has changed; otherwise, false. If this value is true, then any input-layer iterator that you obtained previously becomes invalid.

## Returns

Returns `AcMapIE::ErrCode kErr_OK` if successful. Returns `AcMapIE::ErrCode kErr_DialogNotAvailable` if no driver options dialog box exists for this format. Returns `AcMapIE::ErrCode kErr_Fail` if the dialog box cannot be created or if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: Import Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Performs the import.

```
virtual AcMapIE::ErrCode Import(  
    AcMapIE::CImportResults& results,  
    bool bUseProgressDialog = false  
) = 0;
```

Parameters	Description
results	Output CImportResultsimport results.
bUseProgressDialog	Input true to display AutoCAD Map's progress dialog box during the import, or false to suppress this dialog box.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_OutOfMemory if system resources are insufficient. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

## Remarks

This operation is time-consuming for a large number of entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: ImportPolygonsAsClosedPolylines Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Determines whether polygons are imported as closed polylines or as polygons.

```
virtual bool ImportPolygonsAsClosedPolylines() const = 0;
```

Returns

Returns true if polygons are imported as closed polylines, or false if they are imported as polygons.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: InputLayerIterator Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Retrieves an iterator over the input layers. An input layer refers to a single MIF/MID or Shape file, a level or geometry type in a DGN file, or a geometry type in a Coverage or E00 file. A layer consists of a layer name and a schema for attribute data. You are responsible for determining how this data is imported; for more information, see the AcMapIEInputLayer and AcMapIEColumn classes.

```
virtual AcMapIE::ErrCode InputLayerIterator(  
    AcMapIEInputLayerIterator*& pInputLayers  
) = 0;
```

Parameters

Description

pInputLayers

Output [AcMapIEInputLayerIterator](#) iterator over the input layers. The caller is responsible for deleting this object.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Fail if no data was found. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter::LocationWindowAndOptions Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Determines which location window is used to filter incoming entities.

```
virtual void LocationWindowAndOptions(  
    double& dxMin,  
    double& dxMax,  
    double& dyMin,  
    double& dyMax,  
    AcMapIE::LocationOption& locOption  
) const = 0;
```

Parameters	Description
dxMin	Output minimum X value.
dxMax	Output maximum X value.
dyMin	Output minimum Y value.
dyMax	Output maximum Y value.
locOption	Output LocationOptionlocation name.

Returns

Returns nothing.

Remarks

If the value returned in locOption is AcMapIE::kDontUse, then the window is invalid and its minimum and maximum points are set to 0.0; otherwise, the coordinates are set to the WCS (World Coordinate System) values for the screen size (if locOption is AcMapIE::kUseScreenBoundary), or to the coordinates of the user-defined location window (if locOption is AcMapIE::kUseLocationWindow).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: RemoveReactor Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Removes a reactor from this importer.

```
virtual AcMapIE::ErrCode RemoveReactor(  
    AcMapIEImportReactor * pReactor  
) = 0;
```

Parameters

Description

pReactor

Input [AcMapIEImportReactor](#)import reactor to remove.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if the reactor does not already exist. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: SetAuditClassifiedAfterImport Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Enables or disables classified-property audits after import.

```
virtual void SetAuditClassifiedAfterImport(  
    bool bAudit  
) = 0;
```

Parameters

Description

bAudit

Input true to turn auditing on, or false to turn it off.

Returns

Returns nothing.

Remarks

An audit determines whether each classified-property value is out of range and, if so, sets it to its default value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: SetDriverOptions Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Sets the driver options. Typically, you should retrieve the current driver options from DriverOptions() and make changes as needed by using this function.

```
virtual AcMapIE::ErrCode SetDriverOptions(  
    AcMapIENameValueIterator*& pNameValuePairs,  
    bool& bSchemaHasChanged  
) = 0;
```

Parameters	Description
pNameValuePairs	Input driver options, as <a href="#">AcMapIENameValueIterator</a> name-value pairs.
bSchemaHasChanged	Output true if the schema has changed; otherwise, false. If this value is true, then any input-layer iterator that you obtained previously becomes invalid.

## Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_OutOfMemory if system resources are insufficient. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: SetImportPolygonsAsClosedPolylines Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Sets the manner in which polygons are imported: as closed polylines or as polygons.

```
virtual void SetImportPolygonsAsClosedPolylines(  
    bool bLines  
) = 0;
```

Parameters

Description

bLines

Input true to import polygons as closed polylines, or false to import them as polygons.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImporter Class](#), [AcMapIEImporter Class](#)

AcMapIEImporter:: SetLocationWindowAndOptions Method

[AcMapIEImporter Class](#) | [AcMapIEImporter Class](#)

Sets the location window used to filter incoming entities.

```
virtual AcMapIE::ErrCode SetLocationWindowAndOptions(  
    double dxMin,  
    double dxMax,  
    double dyMin,  
    double dyMax,  
    AcMapIE::LocationOption locOption  
) = 0;
```

Parameters	Description
dxMin	Input minimum X value.
dxMax	Input maximum X value.
dyMin	Input minimum Y value.
dyMax	Input maximum Y value.
locOption	Input LocationOptionlocation name.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_InvalidWindow if window is invalid. Returns AcMapIE::ErrCode kErr\_BadParams if the location option is invalid. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Remarks

If locOption is set to AcMapIE::kDontUse, then all window-location values are discarded. This function fails if dxMin > dxMax or dyMin > dyMax.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImportReactor Class](#), [AcMapIEImportReactor Class](#)  
AcMapIEImportReactor::~~AcMapIEImportReactor Destructor  
[AcMapIEImportReactor Class](#) | [AcMapIEImportReactor Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEImportReactor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImportReactor Class](#), [AcMapIEImportReactor Class](#)  
AcMapIEImportReactor:: AcMapIEImportReactor Constructor  
[AcMapIEImportReactor Class](#) | [AcMapIEImportReactor Class](#)

Constructs an instance of this class.

```
AcMapIEImportReactor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEImportReactor Class](#), [AcMapIEImportReactor Class](#)

AcMapIEImportReactor::RecordImported Method

[AcMapIEImportReactor Class](#) | [AcMapIEImportReactor Class](#)

Invoked after an entity is imported.

```
virtual void RecordImported(  
    AcDbObjectId oId  
);
```

Parameters	Description
oId	Input ID of the imported entity.

Returns

Returns nothing.

Remarks

You can use this function to track imported entities.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEImportReactor Class](#), [AcMapIEImportReactor Class](#)

[AcMapIEImportReactor:: RecordReadyForImport Method](#)

[AcMapIEImportReactor Class](#) | [AcMapIEImportReactor Class](#)

Invoked before an entity is imported.

```
virtual bool RecordReadyForImport(  
    AcDbEntity* pEnt  
);
```

Parameters	Description
pEnt	Input entity.
Returns	

Returns true if the specified entity should be imported; otherwise, returns false.

## Remarks

The entity is not a const, so you can modify it as needed. The return value indicates whether an entity is ready to import. If this function returns false, the entity is discarded and no more reactors in the chain are called. If this function returns true for all reactors in the chain, the entity is added to the database. Only after an entity is added to the database can the entity can be classified or class properties be added to it.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer::~~AcMapIEInputLayer Destructor  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEInputLayer();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer:: ColumnIterator Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves an iterator over a collection of AcMapIEColumninstances.

```
virtual AcMapIEColumnIterator* ColumnIterator() const = 0;
```

Returns

Returns the column iterator. The caller is responsible for deleting this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer:: ImportFromInputLayerOn Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Determines whether this import layer is on (and will be imported).

```
virtual bool ImportFromInputLayerOn() const = 0;
```

Returns

Returns true if this input layer is on (this layer will be imported); otherwise, returns false (this layer will not be imported).

## Remarks

The default is on. In the AutoCAD Map user interface, this setting corresponds to the check box in the Input Layer column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

AcMapIEInputLayer::Name Method

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves the name of this input layer.

```
virtual const ACHAR* Name() const = 0;
```

Returns

Returns the name of this input layer. Do not delete this name.

## Remarks

This name identifies an input layer uniquely and does not change. In the AutoCAD Map user interface, this setting corresponds to the Input Layer column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer::OriginalCoordSys Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves AutoCAD Map's best guess for the coordinate system that the data to be imported uses.

```
virtual const ACHAR* OriginalCoordSys() const = 0;
```

Returns

Returns the name of coordinate system, as a Mentor abbreviation.

## Remarks

This function is not yet implemented in AutoCAD Map and always returns an empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer:: SetImportFromInputLayerOn Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Turns this input layer on (to be imported) or off (not to be imported).

```
virtual void SetImportFromInputLayerOn(  
    bool bOn  
) = 0;
```

Parameters

Description

bOn

Input true to turn this input layer on, or false to turn it off.

Returns

Returns nothing.

Remarks

In the AutoCAD Map user interface, this setting corresponds to the check box in the Input Layer column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

AcMapIEInputLayer:: SetUseForBlockAttributes Method

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Sets whether block attribute values are brought in from incoming attribute data.

```
virtual AcMapIE::ErrCode SetUseForBlockAttributes(  
    bool bGetAttributeFromField  
) = 0;
```

Parameters

Description

bGetAttributeFromField Input true to import attribute values; otherwise, false.

Returns

Returns AcMapIE::ErrCode kErr\_OK if successful. Returns AcMapIE::ErrCode kErr\_Invalid if bGetAttributeFromField is true and the drawing has no blocks. Returns AcMapIE::ErrCode kErr\_Fail if the process failed for some other reason.

Remarks

In the AutoCAD Map user interface, this setting corresponds to the Get Attribute Values From Fields check box in the Point Mapping dialog box. Attributes are brought in when the incoming field name matches the attribute name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)

[AcMapIEInputLayer:: TargetCoordSys Method](#)

[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Retrieves the coordinate system that incoming data is transformed from.

```
virtual const ACHAR* TargetCoordSys() const = 0;
```

Returns

Returns the name of the coordinate system, as a Mentor abbreviation.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Coordinate System column in the Import dialog box properties table.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapIEInputLayer Class](#), [AcMapIEInputLayer Class](#)  
AcMapIEInputLayer:: UseForBlockAttributes Method  
[AcMapIEInputLayer Class](#) | [AcMapIEInputLayer Class](#)

Determines whether block attribute values are brought in from incoming attribute data.

```
virtual bool UseForBlockAttributes() const = 0;
```

Returns

Returns true if attribute values are imported; otherwise, returns false.

## Remarks

In the AutoCAD Map user interface, this setting corresponds to the Get Attribute Values From Fields check box in the Point Mapping dialog box.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)  
AcMapIEInputLayerIterator:: ~AcMapIEInputLayerIterator Destructor  
[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIEInputLayerIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)

AcMapIEInputLayerIterator:: Done Method

[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)

[AcMapIEInputLayerIterator:: Get Method](#)

[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Retrieves the current element in the iteration.

```
virtual bool Get(  
    AcMapIEInputLayer*& pInputLayer  
) = 0;
```

Parameters

Description

pInputLayer

Output pointer to the current AcMapIEInputLayer.

Returns

Returns true if the current element is valid, or false if the iteration has no more elements to retrieve.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)

[AcMapIEInputLayerIterator::Rewind Method](#)

[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIEInputLayerIterator Class](#), [AcMapIEInputLayerIterator Class](#)

[AcMapIEInputLayerIterator:: Step Method](#)

[AcMapIEInputLayerIterator Class](#) | [AcMapIEInputLayerIterator Class](#)

Advances to the next element in the iteration.

```
virtual void Step() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIENameValueIterator Class](#), [AcMapIENameValueIterator Class](#)  
AcMapIENameValueIterator:: ~AcMapIENameValueIterator Destructor  
[AcMapIENameValueIterator Class](#) | [AcMapIENameValueIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapIENameValueIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIENamespaceValueIterator Class](#), [AcMapIENamespaceValueIterator Class](#)

AcMapIENamespaceValueIterator::Clear Method

[AcMapIENamespaceValueIterator Class](#) | [AcMapIENamespaceValueIterator Class](#)

Removes all elements from the collection.

```
virtual void Clear() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapIENamespaceIterator Class](#), [AcMapIENamespaceIterator Class](#)

AcMapIENamespaceIterator:: Done Method

[AcMapIENamespaceIterator Class](#) | [AcMapIENamespaceIterator Class](#)

Determines whether the iterator has reached the end of the collection.

```
virtual bool Done() = 0;
```

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapINameValueIterator Class](#), [AcMapINameValueIterator Class](#)

[AcMapINameValueIterator::Rewind Method](#)

[AcMapINameValueIterator Class](#) | [AcMapINameValueIterator Class](#)

Moves to the first element in the iteration.

```
virtual void Rewind() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapINameValueIterator Class](#), [AcMapINameValueIterator Class](#)

[AcMapINameValueIterator:: Step Method](#)

[AcMapINameValueIterator Class](#) | [AcMapINameValueIterator Class](#)

Advances to the next element in the iteration.

```
virtual void Step() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#)

AcMapMbError:: EMbStatus Enumeration

[AcMapMbError Class](#)

This enum represents the list of defined Map Book errors.

```
enum EMbStatus {
    kMapMbOk = 0,
    kInvalidTemplateName = 1,
    kInvalidLayoutName,
    kTileViewportNotFound,
    kKeyViewportNotFound,
    kLegendViewportNotFound,
    kAdjacentArrowNotFound,
    kTitleBlockNotFound,
    kMultipleTileViewports,
    kMultipleKeyViewports,
    kMultipleLegendViewports,
    kNoDMLegend,
    kInvalidTitleBlockRedefinition,
    kInvalidAdjacentArrowBlockRedefinition,
    kDrawingHasNotBeenSaved = 50,
    kDrawingHasBeenModified,
    kSourceMapNotFound,
    kSourceMapScaleNotFound,
    kSourceMapCannotReset,
    kInvalidSheetSetName = 100,
    kSheetSetNotFound,
    kSheetSetOutOfSynch,
    kTileGenGridRowsOrColsAreZero = 200,
    kTileGenGridVariableTileSizeNotSupported,
    kTileGenGridEmptyExtents,
    kTileGenIncorrectTileSettings,
    kTileNameGenSettingsNull = 300,
    kTileNameGenExpressionFailed,
    kTileNameGenSchemeUnknown,
    kTileNameGenSettingsValueFail,
    kTileNameGenSetTileNameFail,
    kTileNameGenNotGridTiling,
    kTileRelationshipBuilder = 400,
    kTileRelBlderTileAdjacencyFail,
    kTileRelBlderInvalidTable,
    kTileRelBlderUnknowRegion,
    kTileRelBlderRegionFail,
    kXMLFiler = 500,
    kXMLFilerUnrecognizedSettings,
```

```
kTileGeneratorSettings = 600,  
kTileValueNotInitialized,  
kTileNameGeneratorSettings = 700,  
kTileNameGenBadAlphaSequence  
};  
File
```

AcMapMbError.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#)

AcMapMbError:: ErrorClass Enumeration

[AcMapMbError Class](#)

KMapMbClassErrors represents a type of errors associated with Map Book application.

```
enum ErrorClass {  
    kMapMbClassErrors = 0xB9104053  
};  
File
```

AcMapMbError.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

[AcMapMbError:: ~AcMapMbError Destructor](#)

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Destroys an instance of this class.

```
~AcMapMbError(  
    void
```

```
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

[AcMapMbError:: AcMapMbError Constructor](#)

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Constructs an instance of this class.

```
AcMapMbError(  
    void
```

```
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

AcMapMbError:: IsError Method

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Checks if the status code indicates an error.

```
static bool IsError(  
    EMbStatus es  
);
```

Parameters	Description
es	Input error status

Returns

Returns true in the case if es indicates an error, false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

[AcMapMbError:: IsSuccess Method](#)

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Checks if the status code indicates success.

```
static bool IsSuccess(  
    EMbStatus es  
);
```

Parameters	Description
es	Input error status

Returns

Returns true in the case if es indicates success, false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

[AcMapMbError:: IsWarning Method](#)

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Checks if the status code indicates a warning.

```
static bool IsWarning(  
    EMbStatus es  
);
```

Parameters	Description
es	Input error status

Returns

Returns true in the case if es indicates a warning, false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbError Class](#), [AcMapMbError Class](#)

[AcMapMbError:: Message Method](#)

[AcMapMbError Class](#) | [AcMapMbError Class](#)

Loads an error message associated with the error code from resource file.

```
static const ACHAR* Message(  
    EMbStatus es  
);
```

Parameters	Description
es	Input error status

Returns

Returns a pointer to a static buffer with the message loaded.

## Remarks

If there is no specific message, "Unexpected error" is returned. This method is using internal static buffer to load the resource string; it is recommended to either use the returned string immediately or create a local copy.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbError Class](#), [AcMapMbError Class](#)  
AcMapMbError:: PushErrorMessage Method  
[AcMapMbError Class](#) | [AcMapMbError Class](#)

Pushes an error entry to the error stack.

```
static void PushErrorMessage(  
    EMbStatus es  
);
```

Parameters	Description
es	Input error status

Returns

Nothing.

Remarks

The entry includes kMapMbClassErrors as error type, es as an error code and the error message associated with es if any.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#)

AcMapMbMapBook:: EModificationType Enumeration

[AcMapMbMapBook Class](#)

Bit flags indicating to a reactor what kind of modification is taking place.

```
enum EModificationType {  
    kMapBookGeneral = 0,  
    kMapBookTileSetModified = 1,  
    kMapBookRegenerated = 2,  
    kMapBookModifyUndone = 4,  
    kMapBookTileNameChanged = 8,  
    kMapBookTileAdjacencyModified = 16  
};
```

File

AcMapMbMapBook.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: ~AcMapMbMapBook Destructor  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbMapBook();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: AcMapMbMapBook Constructor  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Constructs an instance of this class.

```
AcMapMbMapBook( );
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: dwgInFields Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: dwgOutFields Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: dxfInFields Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: dxfOutFields Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: GetNorthDirection Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrives the north direction.

```
Acad::ErrorStatus GetNorthDirection(  
    double& dAngle  
) const;
```

### Parameters

### Description

dAngle

Output the angle between the vertical and the north direction in degrees; the possible values can be in the interval [0, 360).

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

### Remarks

This is a Map Book property stored with the Sheet or Sheet Subset; if a Sheet Set file has been deleted, it resets the value from the projection or to 0.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

[AcMapMbMapBook:: GetSettings Method](#)

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Gets the Map Sheet Layout Settings used to generate the Map Book. .

```
Acad::ErrorStatus GetSettings(  
    const AcMapMbMapSheetLayoutSettings*& pSettings  
) const;
```

Parameters

Description

pSettings

Output the Map Sheet Layout Settings settings used to generate the Map Book.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

[AcMapMbMapBook:: GetSettings Method](#)

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Gets the Tile Generator settings used to generate the Map Book.

```
Acad::ErrorStatus GetSettings(  
    const AcMapMbTileGeneratorSettings*& pSettings  
) const;
```

Parameters	Description
pSettings	Output the Tile Generator settings used to generate the Map Book.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

[AcMapMbMapBook:: GetSettings Method](#)

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Gets the Tile Name Generator settings used to generate the Map Book.

```
Acad::ErrorStatus GetSettings(  
    const AcMapMbTileNameGeneratorSettings*& pSettings  
) const;
```

Parameters

Description

pSettings

Output the Tile Name Generator settings used to generate the Map Book.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: GetTileCount Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrives the number of tiles.

```
Acad::ErrorStatus GetTileCount(  
    unsigned int& usTileCount  
) const;
```

Parameters	Description
usTileCount	Output tile number.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as TileCount custom property; if a Sheet Set file has been deleted, it retrieves the actual number of tiles in a tile set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: Implementation Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Returns the implementation object.

```
virtual AcMapMbImpMapBook* Implementation();  
Returns
```

Returns the implementation, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: Name Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrieves the name of this Map Book.

```
const ACHAR* Name() const;
```

Returns

Returns the MapBook Name. The function returns NULL if the name is empty. The returned name should not be deleted.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: SetNorthDirection Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Sets the north direction.

```
Acad::ErrorStatus SetNorthDirection(  
    double dAngle  
);
```

Parameters	Description
dAngle	Input the angle between the vertical and the north direction in degrees; the allowed values can be from 0 up to but not equal to 360.

## Returns

Returns Acad::eOk if successful; Acad::eInvalidInput if the value is not in the interval [0, 360) otherwise, returns a different error code.

## Remarks

This is a Map Book property stored with the Sheet or Sheet Subset as NorthDirection custom property; this way it can be seen and even edited in Sheet Set Manager. 0 indicates aligning with the Y axis pointing up, increasing in the clockwise direction.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook::SheetSetFileName Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrieves the file name of the Map Sheet Set associated with this MapBook.

```
const ACHAR* SheetSetFileName() const;
```

Returns

Returns Map Sheet Set file name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)  
AcMapMbMapBook:: SheetSetSubsetName Method  
[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Retrieves the Subset name within the Map Sheet Set if any.

```
const ACHAR* SheetSetSubsetName() const;
```

Returns

Output Subset name within the Map Sheet Set. The Subset name includes the full path where a slash "/" is used as a delimiter, for example:

"Root/Nested\_Subset/Map\_Book\_Subset". If a Map Book is stored as a complete Sheet Set, NULL is returned

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

[AcMapMbMapBook:: ShowGridObjects Method](#)

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Shows/hides grid objects (polylines) associated with the Tile Set.

```
Acad::ErrorStatus ShowGridObjects(  
    bool bToBeShown  
) const;
```

Parameters

Description

bToBeShown

Output the desired visibility status, if false object will be hidden.

Returns

Returns Acad::eOk if successful; returns Acad::eNotApplicable in the case of grid objects being created manually, otherwise, returns a different error code.

Remarks

This functionality is applicable only to grid objects generated automatically; polylines created manually are not affected. By default, grid objects become visible when a Map Book becomes current.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: subClose Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Invoked from within close() before the close actually occurs. The default implementation of this function returns Acad::eOk. See also subClose() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

**virtual** Acad::ErrorStatus subClose();

Returns

Returns Acad::eOk if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: subErase Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Invoked from within erase() before the erase actually occurs. See also subErase() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus subErase(  
    Adesk::Boolean erasing  
);
```

### Parameters

### Description

erasing

Input erasing argument that was passed to the erase() function call that triggered this subErase() call.

### Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook::TileSet Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Gets the Tile Set of the Map Book.

```
AcMapMbTileSet* TileSet() const;
```

Returns

Returns the Map Book Tile Set. Returns NULL in the case of failure.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBook Class](#), [AcMapMbMapBook Class](#)

AcMapMbMapBook:: wblockClone Method

[AcMapMbMapBook Class](#) | [AcMapMbMapBook Class](#)

Grants control of deep clone operations to the object. In the default implementation, the object is cloned and appended to the owner object pOwnerObject. See also wblockClone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus wblockClone(  
    AcRxObject* pOwnerObject,  
    AcDbObject*& pClonedObject,  
    AcDbIdMapping& idMap,  
    Adesk::Boolean isPrimary = true  
) const;
```

Parameters	Description
pOwnerObject	Input object to append the clones to.
pClonedObject	Output the cloned object, or NULL if not cloned.
idMap	Input current ID map.
isPrimary	Input true if this object is primary, or false if it is owned.

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: TILEID Nested Type  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Defines a type for ID's of map book tiles.

```
typedef unsigned long TILEID;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookManager Class](#)

AcMapMbMapBookManager:: EAdjacentTile Enumeration

[AcMapMbMapBookManager Class](#)

Defines a direction for selecting adjacent tiles.

```
enum EAdjacentTile {  
    kAdjacentTop = 0,  
    kAdjacentTopRight,  
    kAdjacentRight,  
    kAdjacentBottomRight,  
    kAdjacentBottom,  
    kAdjacentBottomLeft,  
    kAdjacentLeft,  
    kAdjacentTopLeft  
};
```

File

AcMapMbMapBookManager.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: AddMapBookCreationReactor Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Adds a map book creation reactor to monitor map book creation process such as end of tiling and sheet creation.

```
static bool AddMapBookCreationReactor(  
    AcDbDatabase* pDatabase,  
    AcMapMbMapBookCreationReactor* pReactor  
);
```

Parameters	Description
pDatabase	Input database.
pReactor	
Returns	

Returns true if successful.

## Remarks

This reactor is not currently exposed through the public API and is used for internal purposes only.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager::AddMapBookReactor Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Adds a map-book-management reactor to monitor map book management related activities.

```
static bool AddMapBookReactor(  
    AcDbDatabase* pDatabase,  
    AcMapMbMapBookReactor* pReactor  
);
```

Parameters	Description
pDatabase	Input database.
pReactor	Input reactor.
Returns	

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager::GetCurrent Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Retrieves the current Map Book.

```
static Acad::ErrorStatus GetCurrent(  
    AcDbObjectId& Id,  
    AcDbDatabase* pDatabase  
);
```

Parameters	Description
Id	Output ID of the current Map Book. Null Id corresponds no current Map Book state.
pDatabase	Input pointer to a drawing database.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: GetMapBookForLayout Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Retrieves an Id of a Map Book associated with the given layout if any.

```
static Acad::ErrorStatus GetMapBookForLayout(  
    AcDbObjectId& mapBookId,  
    const AcDbObjectId& layoutId  
);
```

Parameters	Description
mapBookId	Output Id of the Map Book. It is assigned a value only in the case the specified layout is a Map Book layout.
layoutId	Input Id of the AutoCAD Layout object.
Returns	

Returns Acad::eOk if this is a Map Book layout, Acad::eKeyNotFound if this is not a Map Book layout; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: HasMapBookData Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Determines whether a project has map book data.

```
static bool HasMapBookData(  
    AcDbDatabase * pDatabase  
);
```

Parameters	Description
pDatabase	Input database.

Returns

Returns true if the project has map book data.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: mbMapBookManagerImp Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Returns the implementation object.

```
static AcMapMbImpMapBookManager* mbMapBookManagerImp();
```

Returns

Returns the implementation, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: NewIterator Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Returns a new map book dictionary iterator.

```
static AcDbDictionaryIterator* NewIterator(  
    AcDbDatabase* pDatabase  
);
```

Parameters	Description
pDatabase	Input pointer to a drawing database. If the input is NULL this function returns NULL.

## Returns

Returns iterator if Map Book dictionary exists; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: Remove Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Removes the map book with the specified ID.

```
static Acad::ErrorStatus Remove(  
    const AcDbObjectId& Id  
);
```

Parameters

Description

Id

Input ID of the map book to remove.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager::RemoveMapBookCreationReactor Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Removes a map book creation reactor.

```
static bool RemoveMapBookCreationReactor(  
    AcDbDatabase* pDatabase,  
    AcMapMbMapBookCreationReactor* pReactor  
);
```

Parameters	Description
pDatabase	Input database.

pReactor

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager::RemoveMapBookReactor Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Removes a map book reactor from a database.

```
static bool RemoveMapBookReactor(  
    AcDbDatabase* pDatabase,  
    AcMapMbMapBookReactor* pReactor  
);
```

Parameters	Description
pDatabase	Input database.

pReactor

Returns

Returns true if successful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)

[AcMapMbMapBookManager:: SetCurrent Method](#)

[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Sets the current Map Book.

```
static Acad::ErrorStatus SetCurrent(  
    const AcDbObjectId& Id,  
    AcDbDatabase* pDatabase = NULL  
);
```

Parameters	Description
Id	Input ID of the Map Book to set current. Null Id corresponds to current Map Book state. Id should belong to the specified database. Even if a source Display Map or Map scale are missing or a system cannot find the Sheet Set set file associated with the specified Map Book the operation succeeds, however an appropriate error code is returned in these cases.
pDatabase	Input pointer to a drawing database
Returns	

Returns Acad::eOk if successful; returns Acad::eKeyNotFound if DM Map or scale is missing returns Acad::eFileNotFound if proper sheet set not found; returns Acad::eInProcessOfCommitting in the case new Map Book required switching Display Map but there are uncommitted edits; otherwise, returns a different error code.

## Remarks

In the case a Map Book has a Display Map as a source, resets the proper current Display Map. In the case there are uncommitted edits, setting current Display Map and respectively current Map Book fails.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: SetMapBookDMEnvironment Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Sets the Display Management environment such as DM Map and Scale associated with the Map Book.

```
static Acad::ErrorStatus SetMapBookDMEnvironment(  
    const AcMapMbTileGeneratorSettings* pTileGeneratorSettings  
);
```

Parameters	Description
pTileGeneratorSettings	Input settings object for MapBookSource, DMMapName, and DmScaleFactor

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

Uses the tile generator settings object for MapBookSource DMMapName, and DmScaleFactor to set the appropriate environment. These may not be consistent, in which case an error message may be returned and/or messages may be pushed onto the AutoCAD Map Error Stack. The user should check this stack after each call.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookManager Class](#), [AcMapMbMapBookManager Class](#)  
AcMapMbMapBookManager:: SetMapBookEnvironment Method  
[AcMapMbMapBookManager Class](#) | [AcMapMbMapBookManager Class](#)

Sets the environment associated with the Map Book.

```
static Acad::ErrorStatus SetMapBookEnvironment(  
    const AcMapMbMapBook* pMapBook  
);
```

Parameters	Description
pMapBook	Input pointer to an existing MapBook.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This includes opening the related Sheet Set, setting current DM Map and scale etc.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

AcMapMbMapBookReactor:: MapBookAppended Method

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a new map book is appended or after a reactor is attached to a project that has map book(s) defined.

```
virtual void MapBookAppended(  
    AcMapMbMapBook* pMapBook  
);
```

Parameters	Description
pMapBook	Input pointer to the affected MapBook.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

AcMapMbMapBookReactor:: MapBookErased Method

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book is detached or erase operation is undone.

```
virtual void MapBookErased(  
    AcMapMbMapBook* pMapBook,  
    bool bErased  
);
```

Parameters

Description

pMapBook

Input pointer to the affected MapBook.

bErased

Input parameter indicating the kind of operation. Set to true if the MapBook has been erased, and false if the MapBook erase has been reversed.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

AcMapMbMapBookReactor:: MapBookModified Method

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book is detached or erase operation is undone.

```
virtual void MapBookModified(  
    AcMapMbMapBook* pMapBook,  
    unsigned long usFlag  
);
```

Parameters	Description
pMapBook	Input pointer to the affected MapBook.
usFlag	Input bit set where each bit indicates the specific change. The bits are defined by EModificationType enum in AcMapMbMapBook.h.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

[AcMapMbMapBookReactor:: MapBookSetCurrent Method](#)

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a current Map Book gets set or at the very beginning of the Map session.

```
virtual void MapBookSetCurrent(  
    const AcDbObjectId& oldMapBookId,  
    const AcDbObjectId& newMapBookId  
);
```

Parameters	Description
oldMapBookId	Input old current Map Book Id.
newMapBookId	Input new current Map Book Id.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

[AcMapMbMapBookReactor:: MapBookTileModified Method](#)

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book tile is modified.

```
virtual void MapBookTileModified(  
    const AcMapMbTile* pTile,  
    unsigned long usFlag  
);
```

### Parameters

### Description

pTile

Input pointer to the affected Tile.

usFlag

Input bit set where each bit indicates the specific change. The bits are defined by EModificationType enum in AcMapMbMapBook.h.

### Returns

Returns nothing.

### Remarks

Notifications occur when the Tile Name is modified, when Adjacency is modified and when the tile is disabled.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)  
AcMapMbMapBookReactor:: MapBookTileWillBeErased Method  
[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book tile is about to be erased.

```
virtual void MapBookTileWillBeErased(  
    const AcMapMbTile* pTile  
);
```

Parameters	Description
pTile	Input pointer to the affected Tile.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)  
AcMapMbMapBookReactor:: MapBookTreeNodeModified Method  
[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book tree node is modified.

```
virtual void MapBookTreeNodeModified(  
    const AcMapMbTileSet::AcMapTreeNode* pTile,  
    unsigned long usFlag  
);
```

Parameters	Description
usFlag	Input bit set where each bit indicates the specific change. This parameter is not currently used.
pNode	Input pointer to the affected Tree Node.
Returns	

Returns nothing.

## Remarks

Notifications occur when the tree node name is modified.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapBookReactor Class](#), [AcMapMbMapBookReactor Class](#)

[AcMapMbMapBookReactor:: MapBookWillBeErased Method](#)

[AcMapMbMapBookReactor Class](#) | [AcMapMbMapBookReactor Class](#)

Invoked when a map book is about to be erased or erase operation is about to be undone.

```
virtual void MapBookWillBeErased(  
    AcMapMbMapBook* pMapBook,  
    bool bErased  
);
```

Parameters

Description

pMapBook

Input pointer to the affected MapBook.

bErased

Input parameter indicating the kind of operation. Set to true if the MapBook is being erased, and false if the MapBook erase is being reversed.

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: EMapKeyType Enumeration

[AcMapMbMapSheetLayoutSettings Class](#)

Enumerates supported types of a Map Key.

```
enum EMapKeyType {  
    kMapKeyNone,  
    kMapKeyCurrentDrawing,  
    kMapKeyExternalDrawing,  
    kMapKeyLinkedDrawing  
};  
File
```

AcMapMbMapSheetLayoutSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: EMapLegendType Enumeration

[AcMapMbMapSheetLayoutSettings Class](#)

Enumerates supported types of legend

```
enum EMapLegendType {  
    kMapLegendNone,  
    kMapLegendCurrentMap,  
    kMapLegendBox
```

```
};
```

File

AcMapMbMapSheetLayoutSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings::~~AcMapMbMapSheetLayoutSettings  
Destructor

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbMapSheetLayoutSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings::

AcMapMbMapSheetLayoutSettings Constructor

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Constructs an instance of this class.

```
AcMapMbMapSheetLayoutSettings();
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: AdjacentArrowBlockName Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves an adjacent arrow block name.

```
const ACHAR* AdjacentArrowBlockName() const;
```

Returns

Returns an adjacent arrow block name or NULL if has not been specified.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: AllocateMapSheetTemplate Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Allocates an instance of Map Sheet Template class.

**virtual** [AcMapMbMapSheetTemplate](#)\* AllocateMapSheetTemplate() **const**;  
Returns

Returns pointer to the allocated [AcMapMbMapSheetTemplate](#) object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: clone Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings::copyFrom Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: dwgInFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: dwgOutFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: dxfInFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: dxfOutFields Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: GetLegendBox Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a Legend box.

```
Acad::ErrorStatus GetLegendBox(  
    AcDbExtents& extents  
) const;
```

Parameters	Description
extents	Output bounding box.

Returns

Returns Acad::eOk if successful; Acad::eInvalidExtents if the legend box is invalid or has not been initialized, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: Implementation Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Returns the implementation object.

```
virtual AcMapMbImpMapSheetLayoutSettings* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: IncludeAdjacentArrows Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves the flag to include a adjacent arrows to a Map Sheet layout.

```
bool IncludeAdjacentArrows() const;
```

Returns

Returns true if a Map Sheet should include Adjacent Arrows; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: IncludeTitleBlock Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves the flag to include a title block to a Map Sheet layout.

```
bool IncludeTitleBlock() const;
```

Returns

Returns true if a Map Sheet should include a Title block; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: LegendType Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a Legend type.

```
EMapLegendType LegendType() const;
```

Returns

Returns Legend Type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: MapKeyExternalFileName Method  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a drawing file name for the External Drawing Map Key.

```
const ACHAR* MapKeyExternalFileName() const;
```

Returns

Returns file name. The name specifies a complete path name with the .DWT or .DWG extension. Returns the empty string if this path has not been set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: MapKeyLayerPattern Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a pattern to select layers visible within a current drawing Map Key.

```
const ACHAR* MapKeyLayerPattern() const;
```

Returns

Returns pattern to select layers visible within a current drawing Map Key or NULL if has not been specified.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: MapKeyType Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a Map Key type.

```
EMapKeyType MapKeyType() const;
```

Returns

Returns Map Key Type. Returns kMapKeyNone as default.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

[AcMapMbMapSheetLayoutSettings:: SetIncludeAdjacentArrows Method](#)  
[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets the flag to include adjacent arrows to a Map Sheet layout.

```
Acad::ErrorStatus SetIncludeAdjacentArrows(  
    bool bIsOn  
);
```

Parameters	Description
bIsOn	Input whether to include adjacent arrows. If true adjacent arrows get included in a layout; otherwise Map Sheet layout will not include adjacent arrows. The name of the adjacent arrow block gets set via <a href="#">SetAdjacentArrowBlockName()</a> method

## Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetIncludeTitleBlock Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets the flag to include a title block to a Map Sheet layout.

```
Acad::ErrorStatus SetIncludeTitleBlock(  
    bool bIsOn  
);
```

Parameters

Description

bIsOn

Input whether to include a title block. If true a title block gets included in a layout; otherwise Map Sheet layout will not include Title Block object. The name of the title block has to be set via SetTitleBlockName() method in order for SetIncludeTitleBlock() to succeed.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetLegendBox Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a Legend box.

```
Acad::ErrorStatus SetLegendBox(  
    const AcDbExtents& extents  
);
```

Parameters	Description
extents	Input bounding box.

Returns

Returns Acad::eOk if successful; Acad::eInvalidExtents if the legend box is invalid, otherwise, returns a different error code.

## Remarks

In the case of kMapLegendBox type the legend viewport is zoomed to the specified area. Use this area to locate custom legend objects.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetLegendType Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a Legend type.

```
Acad::ErrorStatus SetLegendType(  
    EMapLegendType legendType  
);
```

Parameters

Description

legendType

Input type of legend. In order to specify a kMapLegendBox, a legend box has to be specified via the SetLegendBox() method

Returns

Returns Acad::eOk if successful; Acad::eInvalidExtents if the legend box has not being specified yet, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: SetMapKeyType Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Sets a Map Key type.

```
Acad::ErrorStatus SetMapKeyType(  
    EMapKeyType kMapKeyType  
);
```

Parameters	Description
kMapKeyType	Input type of Map Key.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: TemplateFileName Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a template file name.

```
const ACHAR* TemplateFileName() const;
```

Returns

Returns file name. The name specifies a complete path name with the .DWT or .DWG extension. If this has not been set, the function returns an empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: TemplateLayoutName Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a template layout name.

```
const ACHAR* TemplateLayoutName() const;
```

Returns

Returns layout name. If this has not been set, the function returns an empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetLayoutSettings Class](#), [AcMapMbMapSheetLayoutSettings Class](#)

AcMapMbMapSheetLayoutSettings:: TitleBlockName Method

[AcMapMbMapSheetLayoutSettings Class](#) | [AcMapMbMapSheetLayoutSettings Class](#)

Retrieves a title block name.

```
const ACHAR* TitleBlockName() const;
```

Returns

Returns a title block name or NULL if has not been specified.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetTemplate Class](#)

AcMapMbMapSheetTemplate:: ETemplateElementType Enumeration

[AcMapMbMapSheetTemplate Class](#)

Defines template elements.

```
enum ETemplateElementType {  
    kMapView,  
    kMapKey,  
    kMapLegend,  
    kAdjacentArrow,  
    kTitleBlock  
};  
File
```

AcMapMbMapSheetTemplate.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: ~AcMapMbMapSheetTemplate Destructor  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbMapSheetTemplate();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: AcMapMbMapSheetTemplate Constructor  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Constructs an instance of this class.

```
AcMapMbMapSheetTemplate();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

AcMapMbMapSheetTemplate:: Close Method

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Closes the template drawing file.

```
Acad::ErrorStatus Close();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

AcMapMbMapSheetTemplate::DwgDatabase Method

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Retrieves the drawing database object associated with the Map Sheet template.

```
AcDbDatabase* DwgDatabase() const;
```

Returns

Returns a pointer to a valid drawing database object if successful; otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
[AcMapMbMapSheetTemplate:: GetAdjacentArrow Method](#)  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets the adjacent arrow Id for the specified direction from the layout specified by the settings object.

```
virtual Acad::ErrorStatus GetAdjacentArrow(  
    AcDbObjectId& adjacentArrowId,  
    AcMapMbMapBookManager::EAdjacentTile direction  
) const;
```

Parameters	Description
adjacentArrowId	Output adjacent arrow template element.
direction	Input direction enum indicating which adjacent tile.

### Returns

Returns Acad::eOk if successful; Acad::eKeyNotFound in the case the tile does not have adjacent tile in the specified direction, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: GetAdjacentArrows Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets the adjacent arrow Ids for layout specified by the settings object.

```
virtual Acad::ErrorStatus GetAdjacentArrows(  
    AcDbObjectIdArray& aIds  
) const;
```

Parameters

Description

aIds

Output Ids of the arrow insert objects in the layout.

Returns

Returns Acad::eOk if successful; Acad::eKeyNotFound in the case the layout does not have adjacent arrow placeholder, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
[AcMapMbMapSheetTemplate:: GetArrowDirection Method](#)  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets adjacency direction from the adjacent arrow placeholder.

```
static Acad::ErrorStatus GetArrowDirection(  
    AcMapMbMapBookManager::EAdjacentTile& adjDirection,  
    const AcDbObjectId& objectId  
);
```

Parameters	Description
adjDirection	Output adjacency direction.
objectId	Input Id of the adjacent arrow placeholder that is an insert with the attribute. The value of the attribute should be one of the following nonlocalizable strings "TOP" "TOPRIGHT" "RIGHT" "BOTTOMRIGHT" "BOTTOM" "BOTTOMLEFT" "LEFT" "TOPLEFT"

## Returns

Returns Acad::eOk if successful; Acad::eWrongObjectType in the case the Id is not Id of an adjacent arrow placeholder; Acad::eKeyNotFound if attribute value does not correspond to any of the values above; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: GetElement Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Retrieves the Id of the specified template element.

```
virtual Acad::ErrorStatus GetElement(  
    AcDbObjectId& elementId,  
    ETemplateElementType elementType  
) const;
```

Parameters	Description
elementId	Output element Id.
elementType	Input element type.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

AcMapMbMapSheetTemplate:: GetElement Method

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Returns the object id of the object in the specified layout that is marked as an element of the specified type.

Returns Acad::eOk if successful; Returns Acad::eAmbiguousOutput if there is more than one element of the specified type. Returns Acad::eKeyNotFound if there are no elements of the specified type; otherwise, returns a different error code.

```
static Acad::ErrorStatus GetElement(  
    AcDbObjectId& elementId,  
    const AcDbObjectId& layoutId,  
    ETemplateElementType elementType  
);
```

Parameters	Description
elementId	Output Object id of entity previously marked
layoutId	Input Object id of the layout
elementType	Input Enumerated type identifying the element type
Remarks	

If there is more than one element of the specified type, the function fails.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: GetElement Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Returns an array of object ids of the objects in the specified layout that are marked as an element of the specified type.

Returns Acad::eOk if successful; Returns Acad::eKeyNotFound if there are no entities of the specified type; otherwise, returns a different error code.

```
static Acad::ErrorStatus GetElement(  
    AcDbObjectIdArray& aIds,  
    const AcDbObjectId& layoutId,  
    ETemplateElementType elementType  
);
```

Parameters	Description
aIds	Output Object ids of entity previously marked
layoutId	Input Object id of the layout
elementType	Input Enumerated type identifying the element type

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: GetTileSize Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Gets height and width of tile from the Map viewport.

```
virtual Acad::ErrorStatus GetTileSize(  
    double& dWidth,  
    double& dHeight  
) const;
```

Parameters	Description
dWidth	Output viewport width
dHeight	Output viewport height

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
[AcMapMbMapSheetTemplate:: Implementation Method](#)  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Returns the implementation object.

```
AcMapMbImpMapSheetTemplate* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: IsElement Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Checks if the object specified by the Id is an element of the specified type.

```
static bool IsElement(  
    const AcDbObjectId& Id,  
    ETemplateElementType elementType  
);
```

Parameters	Description
Id	Input Id of the object.
elementType	Input element type.

Returns

Returns true if the object is an element, false otherwise.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

[AcMapMbMapSheetTemplate:: LayoutId Method](#)

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Retrieves current template layout Id.

```
Acad::ErrorStatus LayoutId(  
    AcDbObjectId& layoutId  
) const;
```

Parameters

Description

layoutId

Output Id of the current template layout specified via the settings.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: MarkElement Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Marks the selected template object with XData so that it is considered as a template element of the specified type.

Returns Acad::eOk if successful; otherwise, returns a different error code.

```
static Acad::ErrorStatus MarkElement(  
    const AcDbObjectId& elementId,  
    ETemplateElementType elementType  
);
```

Parameters	Description
elementId	Input Object id of entity to mark
elementType	Input Enumerated type identifying the element type

Remarks

This is a utility that is used to produce Map Book template.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)

AcMapMbMapSheetTemplate:: Open Method

[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Opens the template drawing file for reading.

```
virtual Acad::ErrorStatus Open(  
    const AcMapMbMapSheetLayoutSettings* kpSettings  
);
```

Parameters	Description
kpSettings	Input Map Layot template settings.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

The name of the file and the other settings are defined via the [AcMapMbMapSheetLayoutSettings](#) parameter. In order to change the settings of this object an application has to call Close() method an call Open() with the new settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbMapSheetTemplate Class](#), [AcMapMbMapSheetTemplate Class](#)  
AcMapMbMapSheetTemplate:: UnMarkElement Method  
[AcMapMbMapSheetTemplate Class](#) | [AcMapMbMapSheetTemplate Class](#)

Remove marker XData from the selected template object so that it is considered as a template element any more.

Returns Acad::eOk if successful; otherwise, returns a different error code.

```
static Acad::ErrorStatus UnMarkElement(  
    const AcDbObjectId& elementId  
);
```

Parameters	Description
elementId	Input Object id of entity to clear map book xData
Remarks	

This is a utility that is used to produce Map Book template.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: ~AcMapMbTile Destructor](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTile();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: AcMapMbTile Constructor](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Constructs an instance of this class.

```
AcMapMbTile();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile::AcMapMbTile Constructor](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Constructs an instance of this class.

```
AcMapMbTile(  
    AcMapMbMapBookManager::TILEID tileId,  
    const AcDbExtents& extents,  
    double dViewWidth,  
    double dViewHeight,  
    bool bIsEnabled,  
    const AcDbObjectId& refObjectId,  
    AcMapMbTileSet * pTileSet  
);
```

Parameters	Description
tileId	Input tile id for this tile. Tile Ids should be unique within a drawing.
extents	Input size of the tile in model space units. This value does factor in the scale.
dViewWidth	Input width of the view port in model space units. This value does not include the scale factor.
dViewHeight	Input height of the view port in model space units. This value does not include the scale factor.
refObjectId	Input Object Id of the polygon defining the tile border.
pTileSet	Input tile set pointer to the tile set this tile belongs to.
bEnabled	Input boolean to indicate whether this tile should be enabled or disabled.

## Returns

Returns nothing.

## Remarks

This constructor should be used by any third party tile generator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: AcMapMbTile Constructor](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Copies an instance of this class.

```
AcMapMbTile(  
    const AcMapMbTile& other  
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: AddHyperlink Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Assignes a hyperlink to this tile to the specified adjacent arrow object.

```
Acad::ErrorStatus AddHyperlink(  
    const AcDbObjectId& arrowId,  
    AcMapMbMapBookManager::EAdjacentTile tileDir  
);
```

Parameters	Description
arrowId	Input Id of the arrow object; basically it can be any entity - this method does not check entity type.
kTileDir	Input direction.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

AcMapMbTile:: Center Method

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the center of a tile.

```
virtual const AcGePoint3d& Center() const;
```

Returns

Returns the center of the tile. If the center has not been set, then the point 0,0,0 will be returned.

Remarks

Z values are not significant for this release, but may have some function in the future.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: GetAdjacentTile Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Gets the adjacent tile of the specified direction for the tile.

```
Acad::ErrorStatus GetAdjacentTile(  
    AcMapMbMapBookManager::TILEID& ulTileId,  
    AcMapMbMapBookManager::EAdjacentTile tileDir  
) const;
```

Parameters	Description
ulTileId	Output Id of the adjacent tile.
kTileDir	Input direction.

Returns

Returns Acad::eOk if successful; Acad::eNotApplicable in the case the tile does not have adjacent tile in the specified direction, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: GetAdjacentTile Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Gets the adjacent tile of the specified direction for the tile.

```
Acad::ErrorStatus GetAdjacentTile(  
    AcMapMbTile*& pTile,  
    AcMapMbMapBookManager::EAdjacentTile tileDir  
) const;
```

Parameters	Description
pTile	Output the adjacent tile.
kTileDir	Input direction.

Returns

Returns Acad::eOk if successful; Returns Acad::eNotApplicable in case the tile does not have adjacent tile in the specified direction, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: GetLayoutIdAt Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the layout object id at the index.

```
virtual AcDbObjectId GetLayoutIdAt(  
    int nIndex  
) const;
```

Parameters	Description
nIndex	Input index of the layout we want.

Returns

Returns the object Id of the if successful; otherwise, returns NullId.

## Remarks

Since it is possible for the same tile to appear in several layouts, this will allow the user access to the relevant layouts.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: GetTileSize Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the size of the tile.

```
virtual Acad::ErrorStatus GetTileSize(  
    double& dWidth,  
    double& dHeight  
) const;
```

Parameters	Description
dWidth	Output width of the tile.
dHeight	Output height of the tile.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

## Remarks

This also typically indicates the size of the Reference object, however reference objects may have sizes different from the extents. For instance, if the reference objects are used to set the names of the tiles in a mapbook, they may not be the same size, or even rectangular polylines, but the tiles will be calculated to be the correct size based on the template and map scale. For a grid tiling where the sizes of the tiles are calculated, but there may be an overlap specified, this is the size of the grid elements and does not include the overlap. Get and Set ViewPortSize includes the overlap.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: GetViewPortSize Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Gets the size of the tile as viewed.

```
virtual Acad::ErrorStatus GetViewPortSize(  
    double& dWidth,  
    double& dHeight  
) const;
```

Parameters	Description
dWidth	Output width of the viewed size of the tile.
dHeight	Output height of the viewed size of the tile.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This includes the boundary of the tile plus the overlap.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: Implementation Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the implementation object.

```
AcMapMbImpTile* Implementation() const;
```

Returns

Returns the implementation, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: IsEnabled Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Gets whether the tile is enabled.

```
bool IsEnabled() const;
```

Returns

Returns true if enabled; otherwise, returns false.

## Remarks

Disabled tiles typically do not have sheets created for them, but may be a graphical place holder.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: Name Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the name of the tile.

```
const ACHAR * Name() const;
```

Returns

Returns a const pointer to the string defining the name.

Remarks

Names are set when the tile set is created. Names may be changed.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: OnTileErased Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

This is a notification that every tile except the erased one receives.

```
virtual void OnTileErased(  
    AcMapMbMapBookManager::TILEID ulErasedTileId  
);
```

Parameters	Description
ulTileId	Input Id of the erased tile.

Returns

returns nothing

Remarks

An object should not subscribe for this notification - it will be sent all the time. The default implementation corrects adjacent information if the erased tile is adjacent to this tile.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: RefObject Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the reference object associated with the tile.

```
virtual AcDbObjectId RefObject() const;
```

Returns

Returns AcDbObjectId of the reference object, returns NullId if no reference object has been set.

Remarks

This object will typically be a rectangular closed polyline matching the extents.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: SetAdjacentTile Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Sets the adjacent tile of the specified direction for the tile.

```
Acad::ErrorStatus SetAdjacentTile(  
    AcMapMbMapBookManager::TILEID ulTileId,  
    AcMapMbMapBookManager::EAdjacentTile tileDir  
);
```

Parameters	Description
ulTileId	Input Id of the adjacent tile; if Id is equal to the Id of this tile, the adjacency in the specified direction will be removed.
kTileDir	Input direction.
Returns	

Returns Acad::eOk if successful; Returns Acad::eNotApplicable in case the tile does not have adjacent tile in the specified direction, otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

AcMapMbTile::SheetHandle Method

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the unique handle of the Sheet in the Sheet Set.

```
const ACHAR* SheetHandle() const;
```

Returns

Returns Tile handle

Remarks

This handle string is what IAcSmObjectId::GetHandle() returns.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: StreamIn Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Reads data from a dwg file.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to read this tile data from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: StreamIn Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Reads data from a dxf file.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to read this tile data from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: StreamOut Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Loads data into a dwg file.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to write this tile data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: StreamOut Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Loads data into a dxf file.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to write this tile data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: TileId Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the unique tile Id for this tile.

```
AcMapMbMapBookManager::TILEID TileId() const;
```

Returns

Returns Tile Id

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTile Class](#), [AcMapMbTile Class](#)

[AcMapMbTile:: TileSet Method](#)

[AcMapMbTile Class](#) | [AcMapMbTile Class](#)

Returns the pointer to the Tile Set it belongs to.

```
AcMapMbTileSet* TileSet() const;
```

Returns

Returns object pointer.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGenerator Class](#), [AcMapMbTileGenerator Class](#)  
AcMapMbTileGenerator:: ~AcMapMbTileGenerator Destructor  
[AcMapMbTileGenerator Class](#) | [AcMapMbTileGenerator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileGenerator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGenerator Class](#), [AcMapMbTileGenerator Class](#)  
AcMapMbTileGenerator:: AcMapMbTileGenerator Constructor  
[AcMapMbTileGenerator Class](#) | [AcMapMbTileGenerator Class](#)

Constructs an instance of this class.

```
AcMapMbTileGenerator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGenerator Class](#), [AcMapMbTileGenerator Class](#)

AcMapMbTileGenerator::GenerateTiles Method

[AcMapMbTileGenerator Class](#) | [AcMapMbTileGenerator Class](#)

Creates the tile set from the settings.

```
virtual Acad::ErrorStatus GenerateTiles(  
    AcMapMbTileSet& tileSet,  
    const AcMapMbTileGeneratorSettings& tSettings,  
    double dwidth,  
    double dHeight,  
    AcDbDatabase* pDb  
) = 0;
```

Parameters	Description
tileSet	Output Tile set class.
tSettings	Input Tile settings class.
dWidth	Input width of tiles in model units.
dHeight	Input height of tiles in model units.
pDb	Input Database where Tiles, Tile set, and boundary polylines if any will be created. Typically indicates the current drawing.

## Returns

Returns Acad::eOk if successful; otherwise, returns another value Returns Acad::eInvalidInput if the settings are incorrect for the Generator type. Returns Acad::eAmbiguousInput if global extents are empty or other data problems. Returns Acad::eCreateFailed for Manual tile generator and no reference objects were passed in. Returns Acad::eOutOfMemory for allocate memory or other serious errors.

## Remarks

In addition to the error returns listed below, messages may be added to the Error stack where such messages would be useful to a user in deciding what input settings would be more useful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems

registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#)

AcMapMbTileGeneratorSettings:: EMapBookSource Enumeration

[AcMapMbTileGeneratorSettings Class](#)

This enum defines the different sources for a MapBook.

```
enum EMapBookSource {  
    kMBSourceUnknown = 0,  
    kMBSourceDMMMap,  
    kMBSourceModelSpace,  
    kMBImportOldPlot  
};  
File
```

AcMapMbTileGeneratorSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#)

AcMapMbTileGeneratorSettings:: ETilingScheme Enumeration

[AcMapMbTileGeneratorSettings Class](#)

This enum defines the different tiling schemes we have available.

```
enum ETilingScheme {  
    kTilingSchemeUnknown = 0,  
    kTilingSchemeManual,  
    kTilingSchemeGridArea,  
    kTilingSchemeGridRowCol  
};  
File
```

AcMapMbTileGeneratorSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: ~AcMapMbTileGeneratorSettings Destructor  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileGeneratorSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: AcMapMbTileGeneratorSettings Constructor  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Constructs an instance of this class.

```
AcMapMbTileGeneratorSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: ActiveTilingScheme Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns an enum which specifies the active tiling scheme.

```
ETilingScheme ActiveTilingScheme() const;
```

Returns

Returns tiling scheme or kTilingSchemeUnknown if not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: AllocateTileGenerator Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

This method is used to allocate a tile generator that can be used by the settings object.

```
virtual AcMapMbTileGenerator* AllocateTileGenerator() const;
```

Returns

Returns a pointer to a Tile Generator object or NULL if unsuccessful.

Remarks

The caller is responsible for deleting the object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings::copyFrom Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Copies the contents of an object into the messaged object, if feasible. See also `copyFrom()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters	Description
other	Input pointer to the object to copy from.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)

[AcMapMbTileGeneratorSettings:: DisableEmptyTiles Method](#)

[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the boolean value whether to disable empty tiles.

```
bool DisableEmptyTiles() const;
```

Returns

Returns whether to disable empty tiles

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: DMMapName Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

The pointer points to memory holding the dm map name. It is recommended that if you are going to keep this pointer for any length of time you make a copy of it as the contents can be changed at any time. The user should not attempt to delete this memory. This string can be empty to indicate model space but will never be NULL.

```
const ACHAR* DMMapName() const;
```

Returns

Returns Dm Map Name or empty string to indicate Model space.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: DmScaleFactor Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the scale factor of the DM Map for the Map Book.

```
double DmScaleFactor() const;
```

Returns

Returns scale factor or DBL\_MAX if not set.

## Remarks

This is an identifier only to identify which scale range of a DM map to use for this book.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: dwgInFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: dwgOutFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: dxfInFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters	Description
pFiler	Input filer to use to read the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: dxfOutFields Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: GetIntersectEntitySet Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the selection set of entities intersected.

```
Acad::ErrorStatus GetIntersectEntitySet(  
    AcDbObjectIdArray& SelSet  
) const;
```

Parameters	Description
SelSet	Output id array of selection set.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: Implementation Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the implementation object.

```
virtual AcMapMbImpTileGeneratorSettings* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: IntersectEntities Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the boolean value whether to use a particular set of entities as an intersection set when determining which tiles are empty.

```
bool IntersectEntities() const;
```

Returns

Returns whether to intersect for entities.

## Remarks

If disable empty tiles is false, then the IntersectEntities and IntersectEntitiesSet properties have no effect. If disable empty tiles is true, and intersect entities is true and the Intersect Entities set is non-empty, then this set is used as an intersection set to determine empty tiles. Otherwise the entire drawing is used as the intersection set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)

AcMapMbTileGeneratorSettings:: MapBookSource Method

[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Methods used to store and retrieve the data.

Returns an enum which specifies the Source type.

```
EMapBookSource MapBookSource() const;
```

Returns

Returns Source Type or kMBSourceUnknown if not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: OverlapPercent Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the Overlap percent.

```
double OverlapPercent() const;
```

Returns

Returns overlap percent.

## Remarks

Note that an overlap of 100 indicates that the tile is twice the size of an overlap of 0. In a rectangular grid the tiles will overlap to the midpoint of the adjacent tiles. This convention is used in the User Interface as well.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: ScaleFactor Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns the scale factor for the Map Book.

```
double ScaleFactor() const;
```

Returns

Returns scale factor.

## Remarks

A 1: prefix is assumed so if the scale is 1:200 then this value will return 200. This value is applied to the physical size of the layout main tile viewport to determine tile size.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings::SetActiveTilingScheme Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets an enum which specifies the active tiling scheme.

```
Acad::ErrorStatus SetActiveTilingScheme(  
    ETilingScheme tilingScheme  
);
```

Parameters	Description
tilingScheme	Input tiling scheme.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: SetDisableEmptyTiles Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the boolean value whether to disable empty tiles.

```
Acad::ErrorStatus SetDisableEmptyTiles(  
    bool bDisableEmptyTiles  
);
```

Parameters	Description
bDisableEmptyTiles	Input whether to disable empty tiles.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetDmScaleFactor Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the scale of the DM Map for the Map Book.

```
Acad::ErrorStatus SetDmScaleFactor(  
    const double dDmScaleFactor  
);
```

Parameters	Description
dDmScaleFactor	Input Display Manager scale factor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

This is an identifier only to identify which scale range of a DM map to use for this book.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetIntersectEntities Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the boolean value whether to use the intersection set.

```
Acad::ErrorStatus SetIntersectEntities(  
    const bool bIntersectEntities  
);
```

Parameters	Description
bIntersectEntities	Input Boolean whether to use the intersection set.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetIntersectEntitySet Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the selection set of entities for determining empty tiles.

```
Acad::ErrorStatus SetIntersectEntitySet(  
    const AcDbObjectIdArray& SelSet  
);
```

Parameters	Description
SelSet	Input id array of Selection set.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetMapBookSource Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets an enum which specifies the Source type.

```
Acad::ErrorStatus SetMapBookSource(  
    const EMapBookSource sourceType  
);
```

Parameters	Description
sourceType	Input source type.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
[AcMapMbTileGeneratorSettings:: SetOverlapPercent Method](#)  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the Overlap data member.

```
Acad::ErrorStatus SetOverlapPercent(  
    const double dOverlapPercent  
);
```

Parameters	Description
dOverlapPercent	Input percent of overlap.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: SetScaleFactor Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Sets the scale data member.

```
Acad::ErrorStatus SetScaleFactor(  
    const double dScaleFactor  
);
```

Parameters	Description
dScaleFactor	Input scale factor.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

A 1: prefix is assumed so if the scale is 1:200 then this value should be 200. This value is applied to the physical size of the layout main tile viewport to determine tile size.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettings Class](#), [AcMapMbTileGeneratorSettings Class](#)  
AcMapMbTileGeneratorSettings:: TileLayerName Method  
[AcMapMbTileGeneratorSettings Class](#) | [AcMapMbTileGeneratorSettings Class](#)

Returns a pointer to memory holding the tile layer name.

```
virtual const ACHAR * TileLayerName() const;
```

Returns

Returns tile layer name or an empty string if none has been set.

## Remarks

It is recommended that if you are going to keep this pointer for any length of time you make a copy of it as the contents can be changed at any time. The user should not attempt to delete the memory returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: ~AcMapMbTileGeneratorSettingsArea  
Destructor

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileGeneratorSettingsArea();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea::

AcMapMbTileGeneratorSettingsArea Constructor

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Constructs an instance of this class.

```
AcMapMbTileGeneratorSettingsArea();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: AllocateTileGenerator Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

This method is used to allocate a tile generator that can be used by the settings object.

**virtual** [AcMapMbTileGenerator](#)\* AllocateTileGenerator() **const**;

Returns

Returns a pointer to a Tile Generator object or NULL if unsuccessful.

Remarks

The caller is responsible for deleting the object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: AreaFirstPoint Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Returns the first corner point of the area to tile.

```
virtual const AcGePoint2d& AreaFirstPoint() const;
```

Returns

Returns the first corner point. If this value has not been set, the origin (0, 0) is returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: AreaLastPoint Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Return the second or opposite corner point of the area to tile.

```
virtual const AcGePoint2d& AreaLastPoint() const;
```

Returns

Returns the second or opposite corner point. If this value has not been set, the origin (0, 0) is returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: clone Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea::copyFrom Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: dwgInFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: dwgOutFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: dxfInFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: dxfOutFields Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: GetAreaPoints Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Returns the extent points of the area to tile.

```
virtual Acad::ErrorStatus GetAreaPoints(  
    AcGePoint2d& firstPoint,  
    AcGePoint2d& lastPoint  
) const;
```

Parameters

Description

firstPoint

Output, first corner point

lastPoint

Output, second or opposite corner point

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: GetExtents Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Gets the extents of the area to tile.

```
virtual Acad::ErrorStatus GetExtents(  
    double& dxExtMax,  
    double& dyExtMax,  
    double& dxExtMin,  
    double& dyExtMin  
) const;
```

Parameters	Description
dxExtMax	Output, Holds x value of EXTMAX of area to tile
dyExtMax	Output, Holds y value of EXTMAX of area to tile
dxExtMin	Output, Holds x value of EXTMIN of area to tile
dyExtMin	Output, Holds y value of EXTMIN of area to tile
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: SetAreaFirstPoint Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Sets the first corner point of the area to tile.

```
virtual Acad::ErrorStatus SetAreaFirstPoint(  
    const AcGePoint2d& firstPoint  
);
```

Parameters	Description
firstPoint	Input, first corner point.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: SetAreaLastPoint Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Sets the second or opposite corner point of the area to tile.

```
virtual Acad::ErrorStatus SetAreaLastPoint(  
    const AcGePoint2d& lastPoint  
);
```

Parameters

Description

lastPoint

Input, second or opposite corner

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: SetAreaPoints Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Sets the start point of the area to tile.

```
virtual Acad::ErrorStatus SetAreaPoints(  
    const AcGePoint2d& firstPoint,  
    const AcGePoint2d& lastPoint  
);
```

Parameters

Description

firstPoint

Input, first corner point

lastPoint

Input, second or opposite corner point

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsArea Class](#),

[AcMapMbTileGeneratorSettingsArea Class](#)

AcMapMbTileGeneratorSettingsArea:: SetExtents Method

[AcMapMbTileGeneratorSettingsArea Class](#) |

[AcMapMbTileGeneratorSettingsArea Class](#)

Sets the extents of the area to tile.

```
virtual Acad::ErrorStatus SetExtents(  
    const double dxExtMax,  
    const double dyExtMax,  
    const double dxExtMin,  
    const double dyExtMin  
);
```

Parameters	Description
dxExtMax	Input, Holds x value of EXTMAX of area to tile
dyExtMax	Input, Holds y value of EXTMAX of area to tile
dxExtMin	Input, Holds x value of EXTMIN of area to tile
dyExtMin	Input, Holds y value of EXTMIN of area to tile

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: ~AcMapMbTileGeneratorSettingsGrid

Destructor

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileGeneratorSettingsGrid();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid::

AcMapMbTileGeneratorSettingsGrid Constructor

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Constructs an instance of this class.

```
AcMapMbTileGeneratorSettingsGrid();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: AllocateTileGenerator Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

This method is used to allocate a tile generator that can be used by the settings object.

**virtual** [AcMapMbTileGenerator](#)\* AllocateTileGenerator() **const**;

Returns

Returns a pointer to a Tile Generator object or NULL if unsuccessful.

Remarks

The caller is responsible for deleting the object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: AreaStartPoint Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Return the start point of the area to tile.

```
virtual const AcGePoint2d& AreaStartPoint() const;
```

Returns

Returns the start point.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: clone Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid::copyFrom Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: dwgInFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: dwgOutFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: dxfInFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: dxfOutFields Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid::NumberOfColumns Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Returns the integer value indicating number of columns.

```
int NumberOfColumns() const;
```

Returns

Returns the number of Columns. 0 probably indicates that the value has not been set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid::NumberOfRows Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Returns the integer value indicating number of rows.

```
int NumberOfRows() const;
```

Returns

Returns the number of Rows. 0 probably indicates that the value has not been set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: SetAreaStartPoint Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Set the start point of the area to tile.

```
virtual Acad::ErrorStatus SetAreaStartPoint(  
    const AcGePoint2d& firstPoint  
);
```

Parameters

Description

firstPoint

Lower left point (minimum) to indicate where the area to tile begins.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: SetNumberOfColumns Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Sets the integer value indicating number of columns.

```
Acad::ErrorStatus SetNumberOfColumns(  
    const unsigned int nNumberOfColumns  
);
```

Parameters	Description
nNumberOfColumns	value to store

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsGrid Class](#),

[AcMapMbTileGeneratorSettingsGrid Class](#)

AcMapMbTileGeneratorSettingsGrid:: SetNumberOfRows Method

[AcMapMbTileGeneratorSettingsGrid Class](#) |

[AcMapMbTileGeneratorSettingsGrid Class](#)

Sets the integer value indicating number of rows.

```
Acad::ErrorStatus SetNumberOfRows(  
    const unsigned int nNumberOfRows  
);
```

Parameters	Description
nNumberOfRows	value to store

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual::

~AcMapMbTileGeneratorSettingsManual Destructor

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Destroys an instance of this class.

[~AcMapMbTileGeneratorSettingsManual\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual::

AcMapMbTileGeneratorSettingsManual Constructor

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Constructs an instance of this class.

```
AcMapMbTileGeneratorSettingsManual();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: AllocateTileGenerator Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

This method is used to allocate a tile generator that can be used by the settings object.

**virtual** [AcMapMbTileGenerator](#)\* AllocateTileGenerator() **const**;

Returns

Returns a pointer to a Tile Generator object or NULL if unsuccessful.

Remarks

The caller is responsible for deleting the object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: clone Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual::copyFrom Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: dwgInFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: dwgOutFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters	Description
pFiler	Input filer to use to write the object's data.
Returns	

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: dxfInFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: dxfOutFields Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: GetSelectedTileBoundaries Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Returns the object ids of of entities selected for tile boundaries.

```
Acad::ErrorStatus GetSelectedTileBoundaries(  
    AcDbObjectIdArray& SelSet  
) const;
```

Parameters

Description

SelSet

Output ids of the tile boundaries.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: SetSelectedTileBoundaries Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

Sets the selection set of entities selected.

```
Acad::ErrorStatus SetSelectedTileBoundaries(  
    const AcDbObjectIdArray& SelSet  
);
```

Parameters

Description

SelSet

Input Selection set value to store

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileGeneratorSettingsManual Class](#),

[AcMapMbTileGeneratorSettingsManual Class](#)

AcMapMbTileGeneratorSettingsManual:: TileLayerName Method

[AcMapMbTileGeneratorSettingsManual Class](#) |

[AcMapMbTileGeneratorSettingsManual Class](#)

This is overridden from the base class since layer name is not appropriate for manual selection.

```
virtual const ACHAR * TileLayerName() const;
```

Returns

Returns the empty string.

Remarks

This function returns the empty string.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGenerator Class](#), [AcMapMbTileNameGenerator Class](#)  
[AcMapMbTileNameGenerator:: ~AcMapMbTileNameGenerator Destructor](#)  
[AcMapMbTileNameGenerator Class](#) | [AcMapMbTileNameGenerator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileNameGenerator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGenerator Class](#), [AcMapMbTileNameGenerator Class](#)  
[AcMapMbTileNameGenerator:: AcMapMbTileNameGenerator Constructor](#)  
[AcMapMbTileNameGenerator Class](#) | [AcMapMbTileNameGenerator Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGenerator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGenerator Class](#), [AcMapMbTileNameGenerator Class](#)

[AcMapMbTileNameGenerator::GenerateTileNames Method](#)

[AcMapMbTileNameGenerator Class](#) | [AcMapMbTileNameGenerator Class](#)

Iterates through tile set and sets the names of individual tiles.

```
virtual Acad::ErrorStatus GenerateTileNames(  
    const AcMapMbTileNameGeneratorSettings& tSettings,  
    AcMapMbTileSet& TileSet  
) = 0;
```

Parameters	Description
tSettings	Input Tile Name settings class.
tileSet	Output Tile set class.

Returns

Returns Acad::eOk if successful; otherwise, returns another value Returns Acad::eInvalidInput if the settings are incorrect for the Generator type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorData Class](#), [AcMapMbTileNameGeneratorData Class](#)

AcMapMbTileNameGeneratorData:: ~AcMapMbTileNameGeneratorData  
Destructor

[AcMapMbTileNameGeneratorData Class](#) | [AcMapMbTileNameGeneratorData Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileNameGeneratorData();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorData Class](#), [AcMapMbTileNameGeneratorData Class](#)

AcMapMbTileNameGeneratorData::

AcMapMbTileNameGeneratorData Constructor

[AcMapMbTileNameGeneratorData Class](#) | [AcMapMbTileNameGeneratorData Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorData();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorData Class](#), [AcMapMbTileNameGeneratorData Class](#)

AcMapMbTileNameGeneratorData::GenerateTileNames Method

[AcMapMbTileNameGeneratorData Class](#) | [AcMapMbTileNameGeneratorData Class](#)

Iterates through tile set and sets the names of individual tiles.

```
virtual Acad::ErrorStatus GenerateTileNames(  
    const AcMapMbTileNameGeneratorSettings& tSettings,  
    AcMapMbTileSet& TileSet  
);
```

Parameters	Description
tSettings	Input Tile Name settings class.
tileSet	Output Tile set class.

## Returns

Returns Acad::eOk if successful; otherwise, returns another value Returns Acad::eInvalidInput if the settings are incorrect for the Generator type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorGrid Class](#), [AcMapMbTileNameGeneratorGrid Class](#)

AcMapMbTileNameGeneratorGrid:: ~AcMapMbTileNameGeneratorGrid

Destructor

[AcMapMbTileNameGeneratorGrid Class](#) | [AcMapMbTileNameGeneratorGrid Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileNameGeneratorGrid();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorGrid Class](#), [AcMapMbTileNameGeneratorGrid Class](#)

AcMapMbTileNameGeneratorGrid::

AcMapMbTileNameGeneratorGrid Constructor

[AcMapMbTileNameGeneratorGrid Class](#) | [AcMapMbTileNameGeneratorGrid Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorGrid();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorGrid Class](#), [AcMapMbTileNameGeneratorGrid Class](#)

AcMapMbTileNameGeneratorGrid::GenerateTileNames Method

[AcMapMbTileNameGeneratorGrid Class](#) | [AcMapMbTileNameGeneratorGrid Class](#)

Iterates through tile set and sets the names of individual tiles.

```
virtual Acad::ErrorStatus GenerateTileNames(  
    const AcMapMbTileNameGeneratorSettings& tSettings,  
    AcMapMbTileSet& TileSet  
);
```

Parameters	Description
tSettings	Input Tile Name settings class.
tileSet	Output Tile set class.

Returns

Returns Acad::eOk if successful; otherwise, returns another value Returns Acad::eInvalidInput if the settings are incorrect for the Generator type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: EIndexOrientation Enumeration

[AcMapMbTileNameGeneratorIndexer Class](#)

This is record AcMapMbTileNameGeneratorIndexer::EIndexOrientation.

```
enum EIndexOrientation {  
    kOrientUnknown = 0,  
    kOrientRow,  
    kOrientColumn  
};
```

File

AcMapMbTileNameGeneratorSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: ~AcMapMbTileNameGeneratorIndexer

Destructor

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Destroys an instance of this class.

```
~AcMapMbTileNameGeneratorIndexer();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer::

AcMapMbTileNameGeneratorIndexer Constructor

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorIndexer(  
    EIndexOrientation Orient = kOrientUnknown,  
    bool bIsInverted = false,  
    bool bIsNumeric = true,  
    long lIndex = 1,  
    unsigned short uIncrement = 1  
);
```

Parameters	Description
Orient	Input orientation. Tells the user what kind of index this is.
bIsInverted	Input whether the index starts at the front (top) or back (bottom)
bIsNumeric	Input whether the index is numeric or alphabetic.
lIndex	Input starting index value.
uIncrement	Input increment between values of the index.
Returns	

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: Increment Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Retrieves the increment value for the index.

```
unsigned short Increment() const;
```

Returns

Returns the increment value.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: Index Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Retrieves the starting Index.

```
int Index() const;
```

Returns

Returns the starting Index as a number.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: IsInverted Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Retrieves whether the indexing should start at the top or front or at the back or bottom.

**bool** IsInverted() **const**;

Returns

Returns indexing direction.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: IsNumeric Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Retrieves whether the indexing is numeric or alphabetic.

```
bool IsNumeric() const;
```

Returns

Returns if Indexing is numeric.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: Orientation Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

Retrieves the orientation or type of Index.

```
EIndexOrientation Orientation() const;
```

Returns

Returns the orientation.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamIn Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDwgFiler* pFiler  
);
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamIn Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamIn, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDxfFiler* pFiler  
);
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamOut Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDwgFiler* pFiler  
) const;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorIndexer Class](#),

[AcMapMbTileNameGeneratorIndexer Class](#)

AcMapMbTileNameGeneratorIndexer:: StreamOut Method

[AcMapMbTileNameGeneratorIndexer Class](#) |

[AcMapMbTileNameGeneratorIndexer Class](#)

This is StreamOut, a member of class AcMapMbTileNameGeneratorIndexer.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDxfFiler* pFiler  
) const;
```

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSequence Class](#),

[AcMapMbTileNameGeneratorSequence Class](#)

AcMapMbTileNameGeneratorSequence::

~AcMapMbTileNameGeneratorSequence Destructor

[AcMapMbTileNameGeneratorSequence Class](#) |

[AcMapMbTileNameGeneratorSequence Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileNameGeneratorSequence();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSequence Class](#),

[AcMapMbTileNameGeneratorSequence Class](#)

AcMapMbTileNameGeneratorSequence::

AcMapMbTileNameGeneratorSequence Constructor

[AcMapMbTileNameGeneratorSequence Class](#) |

[AcMapMbTileNameGeneratorSequence Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorSequence();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSequence Class](#),

[AcMapMbTileNameGeneratorSequence Class](#)

AcMapMbTileNameGeneratorSequence::GenerateTileNames Method

[AcMapMbTileNameGeneratorSequence Class](#) |

[AcMapMbTileNameGeneratorSequence Class](#)

Iterates through tile set and sets the names of individual tiles.

```
virtual Acad::ErrorStatus GenerateTileNames(  
    const AcMapMbTileNameGeneratorSettings& tSettings,  
    AcMapMbTileSet& tileSet  
);
```

Parameters

Description

tSettings

Input Tile Name settings class.

tileSet

Output Tile set class.

Returns

Returns Acad::eOk if successful; otherwise, returns another value Returns Acad::eInvalidInput if the settings are incorrect for the Generator type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: ETileNamingScheme Enumeration

[AcMapMbTileNameGeneratorSettings Class](#)

&nbsp;

```
enum ETileNamingScheme {  
    kTileNamingSchemeUnknown = 0,  
    kTileNamingSchemeGridColRow,  
    kTileNamingSchemeGridSequential,  
    kTileNamingSchemeSequential,  
    kTileNamingSchemeData  
};  
File
```

AcMapMbTileNameGeneratorSettings.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: ~AcMapMbTileNameGeneratorSettings  
Destructor

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileNameGeneratorSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings::

AcMapMbTileNameGeneratorSettings Constructor

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorSettings();
```

Returns

Returns nothing.

## Remarks

The user should never directly create an instance of this class, only the derived classes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: AllocateTileNameGenerator Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Returns the associated Name Generator.

```
virtual AcMapMbTileNameGenerator* AllocateTileNameGenerator() const;
```

Returns

Returns Name Generator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings::copyFrom Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: CountDisabledTiles Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Returns the count disable tiles data mamber.

```
bool CountDisabledTiles() const;
```

Returns

Returns whether disabled tiles should be counted.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: dwgInFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: dwgOutFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: dxfInFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: dxfOutFields Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: Implementation Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Returns the implementation object.

```
virtual AcMapMbImpTileNameGeneratorSettings* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

## Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: SetCountDisabledTiles Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Sets the count disable tiles data member.

```
Acad::ErrorStatus SetCountDisabledTiles(  
    const bool bCountDisabledTiles  
);
```

Parameters

Description

bCountDisabledTiles    Input to specify how to handle disabled tiles.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

When true, disabled tiles are named, and the index is incremented. When false, disabled tiles are skipped and the index is not incremented.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: SetTileNamingScheme Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Returns an enum which specifies the tile naming scheme.

```
Acad::ErrorStatus SetTileNamingScheme(  
    const ETileNamingScheme tileNamingScheme  
);
```

Parameters	Description
tileNamingScheme	variable to set enum value

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettings Class](#),

[AcMapMbTileNameGeneratorSettings Class](#)

AcMapMbTileNameGeneratorSettings:: TileNamingScheme Method

[AcMapMbTileNameGeneratorSettings Class](#) |

[AcMapMbTileNameGeneratorSettings Class](#)

Returns an enum which specifies the tile naming scheme.

```
ETileNamingScheme TileNamingScheme() const;
```

Returns

Returns tile naming scheme.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData::

~AcMapMbTileNameGeneratorSettingsData Destructor

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Destroys an instance of this class.

**virtual** `~AcMapMbTileNameGeneratorSettingsData();`

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData::

AcMapMbTileNameGeneratorSettingsData Constructor

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorSettingsData();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData::

AllocateTileNameGenerator Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Returns the associated Name Generator.

```
virtual AcMapMbTileNameGenerator* AllocateTileNameGenerator() const;
```

Returns

Returns Name Generator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: clone Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData::copyFrom Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: dwgInFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: dwgOutFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: dxfInFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: dxfOutFields Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsData Class](#),

[AcMapMbTileNameGeneratorSettingsData Class](#)

AcMapMbTileNameGeneratorSettingsData:: Expression Method

[AcMapMbTileNameGeneratorSettingsData Class](#) |

[AcMapMbTileNameGeneratorSettingsData Class](#)

Retrieves the expression to be applied to the reference object.

```
const ACHAR * Expression() const;
```

Returns

Returns the expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: EGridNamingOrder Enumeration

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

This enum indicates the naming order for the grid.

```
enum EGridNamingOrder {
    kGridNamingOrderUnknown = 0,
    kGridNamingOrderLRTB,
    kGridNamingOrderLRBT,
    kGridNamingOrderRLTB,
    kGridNamingOrderRLBT,
    kGridNamingOrderTBLR,
    kGridNamingOrderTBRL,
    kGridNamingOrderBTLR,
    kGridNamingOrderBTRL
};
File
```

AcMapMbTileNameGeneratorSettingsGrid.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid::

~AcMapMbTileNameGeneratorSettingsGrid Destructor

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Destroys an instance of this class.

**virtual** `~AcMapMbTileNameGeneratorSettingsGrid()`;

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid::

AcMapMbTileNameGeneratorSettingsGrid Constructor

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorSettingsGrid();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid::

AllocateTileNameGenerator Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Returns the associated Name Generator.

```
virtual AcMapMbTileNameGenerator* AllocateTileNameGenerator() const;
```

Returns

Returns Name Generator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: clone Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid::copyFrom Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: dwgInFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: dwgOutFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: dxfInFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object read its data. See also `dxfInFields()` in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns `Acad::eOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: dxfOutFields Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: GetPrimaryIndex Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Returns the primary indexer.

```
Acad::ErrorStatus GetPrimaryIndex(  
    AcMapMbTileNameGeneratorIndexer& primaryIndex  
) const;
```

Parameters

Description

rowIndex

Class encapsulating Index orientation and incrementing

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: GetSecondaryIndex Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Returns the column index data member.

```
Acad::ErrorStatus GetSecondaryIndex(  
    AcMapMbTileNameGeneratorIndexer& secondaryIndex  
) const;
```

Parameters	Description
nColumnIndex	Variable to hold value set

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: Separator Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Returns the separator data member.

```
const ACHAR* Separator() const;
```

Parameters

Description

sSeparator

Variable to hold value set

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: SetPrimaryIndex Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Sets the primary indexer.

```
Acad::ErrorStatus SetPrimaryIndex(  
    const AcMapMbTileNameGeneratorIndexer& primaryIndex  
);
```

Parameters

Description

primaryIndex

Class encapsulating Index orientation and incrementing

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsGrid Class](#),

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

AcMapMbTileNameGeneratorSettingsGrid:: SetSecondaryIndex Method

[AcMapMbTileNameGeneratorSettingsGrid Class](#) |

[AcMapMbTileNameGeneratorSettingsGrid Class](#)

Sets the primary indexer.

```
Acad::ErrorStatus SetSecondaryIndex(  
    const AcMapMbTileNameGeneratorIndexer& secondaryIndex  
);
```

Parameters	Description
nColumnIndex	Variable to hold value set

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence::

~AcMapMbTileNameGeneratorSettingsSequence Destructor

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Destroys an instance of this class.

**virtual** [~AcMapMbTileNameGeneratorSettingsSequence\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence::

AcMapMbTileNameGeneratorSettingsSequence Constructor

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Constructs an instance of this class.

```
AcMapMbTileNameGeneratorSettingsSequence();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence::

AllocateTileNameGenerator Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Returns the associated Name Generator.

```
virtual AcMapMbTileNameGenerator* AllocateTileNameGenerator() const;
```

Returns

Returns Name Generator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: clone Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Clones the object. See also clone() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual AcRxObject* clone() const;
```

Returns

Returns a pointer to the clone, or NULL if cloning was unsuccessful.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence::copyFrom Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Copies the contents of an object into the messaged object, if feasible. See also copyFrom() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus copyFrom(  
    const AcRxObject* other  
);
```

Parameters

Description

other

Input pointer to the object to copy from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: dwgInFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object read its data. See also dwgInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgInFields(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: dwgOutFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object write its data. See also dwgOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dwgOutFields(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: dxfInFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object read its data. See also dxfInFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfInFields(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer to use to read the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: dxfOutFields Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Lets this object write its data. See also dxfOutFields() in the AutoCAD ObjectARX Developer's Guide. This overridden function is called by the system as needed; it is unlikely that you will need to call it directly.

```
virtual Acad::ErrorStatus dxfOutFields(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer to use to write the object's data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: GetIndex Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Retrieves the sequence indexer.

```
Acad::ErrorStatus GetIndex(  
    AcMapMbTileNameGeneratorIndexer& sequenceIndex  
) const;
```

Parameters	Description
sequenceIndex	Output class encapsulating Index orientation and incrementing

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileNameGeneratorSettingsSequence Class](#),

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

AcMapMbTileNameGeneratorSettingsSequence:: SetIndex Method

[AcMapMbTileNameGeneratorSettingsSequence Class](#) |

[AcMapMbTileNameGeneratorSettingsSequence Class](#)

Sets the primary indexer.

```
Acad::ErrorStatus SetIndex(  
    const AcMapMbTileNameGeneratorIndexer& sequenceIndex  
);
```

Parameters	Description
sequenceIndex	Output class encapsulating Index orientation and incrementing

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet::EMbNameSchema Enumeration

[AcMapMbTileSet Class](#)

This is record AcMapMbTileSet::EMbNameSchema.

```
enum EMbNameSchema {  
    kNumericSequence,  
    kAlphaSequence,  
    kByDataSequence,  
    kGrid,  
    kLastSchema  
};  
File
```

AcMapMbTileSet.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet:: EMbTreeType Enumeration

[AcMapMbTileSet Class](#)

This is record AcMapMbTileSet::EMbTreeType.

```
enum EMbTreeType {  
    kSequence,  
    kAlpha,  
    kRowCol,  
    kColRow,  
    kCustom  
};  
File
```

AcMapMbTileSet.h

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet:: AcMapContainer Class

[AcMapMbTileSet Class](#)

Alternate interface for tree.

```
class AcMapContainer;  
File
```

AcMapMbTileSet.h

Remarks

Since this tree is modeling a directory structure, some nodes are containers and some are tiles. Also some containers contain other containers. This alternate interface handles this and may be more easily understandable. It also allows the user to look at grids either bottom to top, or top to bottom; right to left, or left to right.

☐ Methods

- ◆ [~AcMapContainer](#) Destroys an instance of this class.
- ◆ [AcMapContainer](#) Constructs an instance of this class.
- [GetContainerIterator](#) Returns the container iterator for this container.
- [GetTileIterator](#) Returns the tile iterator for this container.
- [Name](#) Sets the Node name for the node associated with this container.
- [SetName](#) Sets a node name for the node associated with this container.
- [TreeNode](#) Returns the node associated with this container.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet:: AcMapContainerIterator Class

[AcMapMbTileSet Class](#)

This is class AcMapMbTileSet::AcMapContainerIterator.

```
class AcMapContainerIterator;
```

File

AcMapMbTileSet.h

☐ Methods



[~AcMapContainerIterator](#) Destroys an instance of this class.



[AcMapContainerIterator](#) Constructs an instance of this class.

[Done](#) Tells the user if we are done.

[Get](#) Returns the next node in the chain as a container.

[Length](#) Returns the number of containers in the iterator.

[Rewind](#) Resets the iterator.

[Step](#) Moves to the next node in the chain as a container.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet:: AcMapTileIterator Class

[AcMapMbTileSet Class](#)

This is class AcMapMbTileSet::AcMapTileIterator.

```
class AcMapTileIterator;
```

File

AcMapMbTileSet.h

☐ Methods

⇒ [~AcMapTileIterator](#)

Destroys an instance of this class.

⇒ [AcMapTileIterator](#)

Constructs an instance of this class.

[Done](#)

Tells the user if we are done.

[Get](#)

Returns the current tile in the chain.

[GetNode](#)

Returns the node associated with the current tile in the chain.

[Length](#)

Returns the number of containers in the iterator.

[Rewind](#)

Resets the iterator.

[SetCurrentSecondaryIndexName](#)

Sets a node name for the node associated with this tile.

[Step](#)

Moves to the next tile in the chain.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#)

AcMapMbTileSet:: AcMapTreeNode Class

[AcMapMbTileSet Class](#)

Class AcMapTreeNode models the most general structure for an organization of tiles.

```
class AcMapTreeNode;
```

File

AcMapMbTileSet.h

Remarks

It resembles a directory structure. Every node contains either a tile or a child. The root node contains only a child. Thus a node may be thought of as either a container or a tile. The current organizations are either a 2 dimensional grid, or a sequence of tiles. For a grid, the root node contains one child, this child has siblings for each row in the grid, or each column in the grid, depending on what the user asked for. If the user asked for Row, Column then each sibling is a row container, that is each sibling has no tiles but has a child. The siblings of that child correspond to the columns, and each one has a tile pointer but no children. For a sequence, the root node contains one child. The child and all its siblings contain tiles. If the naming scheme is numeric, then its appropriate to ask for the sequence. If alphabetic, then its more appropriate to ask for an alphabetized list.

☐ Methods

◆ [~AcMapTreeNode](#) Destroys an instance of this class.

◆ [AcMapTreeNode](#) Constructs an instance of this class.

[AddChild](#) Sets a child for this node.

[AddSibling](#) Sets a sibling for this node.

[Child](#) Returns the child of this node.

[ConstChild](#) Returns the const child of this node.

[ConstSibling](#) Returns the const sibling of this node.

[ConstTile](#) Returns the const tile of this node.

[Name](#) Sets the Node name.

<a href="#">SetName</a>	Sets a node name.
<a href="#">SetSibling</a>	Sets a sibling for this node.
<a href="#">SetTitle</a>	Sets the tile for this node.
<a href="#">Sibling</a>	Returns the sibling of this node.
<a href="#">Tile</a>	Returns the tile of this node.
<a href="#">TileSet</a>	Returns the owner tile set of this node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: ~AcMapMbTileSet Destructor](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMbTileSet();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: AcMapMbTileSet Constructor](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Constructs an instance of this class.

```
AcMapMbTileSet(  
    AcDbDatabase * pDb
```

```
);
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: AddTiles Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Sets the new tiles into the tile set.

```
virtual Acad::ErrorStatus AddTiles(  
    AcMapTreeNode* pTree,  
    EMbTreeType treeType  
);
```

Parameters	Description
pTree	Input tree of tiles and nodes. The structure of this tree is based on the tree nodes, children, and siblings described above.
treeType	Input type of tree of tiles and nodes. A kSequential or kAlpha tree should be flat, while a kRowCol or kColRow tree should be a grid with depth 2. kCustom tree may have a more complex tree structure. A kRowCol tree may be viewed as kColRow and vice versa All trees may be viewed as sequential with some ordering. kCustom trees may not be viewed as kRowCol or kColRow, but may be viewed as sequential. A breadth first search order is assumed.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: BuildAdjacentInfo Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Generate adjacency information and assigns it to every tile.

```
Acad::ErrorStatus BuildAdjacentInfo();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: ColumnNumber Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the number of columns in a grid tiling.

```
virtual unsigned int ColumnNumber() const;
```

Returns

Returns the number of columns in a grid tiling. Returns 0 if this is not a grid tiling.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: ConstContainer Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the root of the tree organized according to the tree type as a container.

```
virtual const AcMapContainer* ConstContainer(  
    EMbTreeType howToGet  
) const;
```

Parameters

Description

howToGet

Input type of tree to return.

Returns

Returns the const container if successful; otherwise, returns NULL to indicate the tree has not been set, or the tree type is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: ConstTree Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the root of the tree organized according to the tree type.

```
virtual const AcMapTreeNode* ConstTree(  
    EMbTreeType howToGet  
) const;
```

Parameters

Description

howToGet

Input type of tree to return.

Returns

Returns the const root tree node if successful; otherwise, returns NULL to indicate the tree has not been set, or the tree type is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: CreateNode Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Creates a new node for the tree.

```
virtual AcMapTreeNode* CreateNode(  
    bool bRootNode  
);
```

Parameters

Description

bRootNode

Input boolean indicating whether this node is the root or a branch on the tree.

Returns

Returns a tree node if successful; otherwise, returns NULL

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: Database Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the pointer to the drawing database the tile set is associated with.

```
AcDbDatabase* Database() const;
```

Returns

Returns a valid database pointer if succeeded, NULL if the object is not initialized yet.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: Erase Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Removes the specified tile from the tile set.

```
Acad::ErrorStatus Erase(  
    AcMapMbMapBookManager::TILEID id,  
    bool bEraseSheet  
);
```

Parameters	Description
id	Input tile id.
bEraseSheet	Input flag to erase the corresponding sheet.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Erases all the corresponding objects including a Sheet.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: Find Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the tile corresponding to the given tile id if it exists.

```
virtual AcMapMbTile* Find(  
    AcMapMbMapBookManager::TILEID id  
);
```

Parameters	Description
id	Input tile id.

Returns

Returns the const tile if successful; otherwise, returns a NULL.

## Remarks

If the corresponding Map Book is open for read, the client can call only the constant methods; nonconstant methods require the Map Book being opened for write. Use the retrieved pointer only while the Map Book object is opened.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: GetExtents Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the total extents of the view ports of this tile set.

```
virtual Acad::ErrorStatus GetExtents(  
    AcDbExtents& extents  
) const;
```

Parameters	Description
extents	Output extents.

Returns

Returns true if successful; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: Implementation Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the implementation object.

```
virtual AcMapMbImpTileSet* Implementation();
```

Returns

Returns the implementation, or NULL if unsuccessful.

Remarks

This function is for internal use only; do not call it directly.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: LayerId Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

When a tile set gets generated (except manual), it creates polylines per every tile and puts them to the specified layer.

```
AcDbObjectId LayerId() const;
```

Returns

Id of the AcDbLayerTableRecordObject or AcDbObjectId::kNull in the case of manual grid.

Remarks

This method retrieves Id of this layer.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: MapBook Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the Id of the AcMapMbMapBook object it belongs to.

```
AcMapMbMapBook* MapBook() const;
```

Returns

Returns a valid Id if succeeded, NULL Id if the object is not initialized yet or is not in the database.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: NamingScheme Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the way or schema used when naming the tiles.

```
virtual EMbNameSchema NamingScheme() const;
```

Returns

Returns an enum indicating how the tiles were named.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: RowNumber Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the number of rows in a grid tiling.

```
virtual unsigned int RowNumber() const;
```

Returns

Returns the number of rows in a grid tiling. Returns 0 if this is not a grid tiling.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: SetAdjacentProperties Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Iterates a Tile Set and sets adjacent properties for the corresponding Sheets.

```
Acad::ErrorStatus SetAdjacentProperties();
```

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: StreamIn Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Reads data from a dwg file.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDwgFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to read this tile data from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: StreamIn Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Reads data from a dxf file.

```
virtual Acad::ErrorStatus StreamIn(  
    AcDbDxfFiler* pFiler  
);
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to read this tile data from.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: StreamOut Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Loads data into a dwg file.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDwgFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to write this tile data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: StreamOut Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Loads data into a dxf file.

```
virtual Acad::ErrorStatus StreamOut(  
    AcDbDxfFiler* pFiler  
) const;
```

Parameters

Description

pFiler

Input filer. The filer should be open and ready to write this tile data.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet:: TileNumber Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the total number of tiles in the set.

```
virtual unsigned int TileNumber() const;
```

Returns

Returns the total number of tiles in the set

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

[AcMapMbTileSet::TilingScheme Method](#)

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the schema or organization used when creating the tiles.

```
virtual EMbTreeType TilingScheme() const;
```

Returns

Returns an enum indicating how the tiles are organized.

## Remarks

Once the tiles are created, they may be viewed in the same format they were created, or in a simpler way. For instance, any tree may be viewed as a sequential tree. Any RowCol tree may be viewed as a ColRow tree and vice versa. The following trees and containers are available based on the tree type given in the AddTiles method.

AddTiles ConstContainer, ConstTree or Tree kSequence: kSequence, kAlpha  
kAlpha: kSequence, kAlpha kRowCol: kRowCol, kColRow, kSequence, kAlpha  
kColRow: kRowCol, kColRow, kSequence, kAlpha kCustom: kCustom,  
kSequence, kAlpha.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet Class](#), [AcMapMbTileSet Class](#)

AcMapMbTileSet:: Tree Method

[AcMapMbTileSet Class](#) | [AcMapMbTileSet Class](#)

Returns the root of the tree organized according to the tree type.

```
virtual AcMapTreeNode* Tree(  
    EMbTreeType howToGet  
);
```

Parameters	Description
howToGet	Input type of tree to return.

Returns

Returns the root tree node if successful; otherwise, returns NULL to indicate the tree has not been set, or the tree type is invalid.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).



## map\_dwgbreakobj

### [Map Boundary Functions](#)

---

Breaks linear objects where they cross boundaries.

int

```
map_dwgbreakobj(  
    ads_name sscut,  
    ads_name boundary,  
    short skiptopo,  
    short keepod,  
    ads_name ssreturn);
```

Returns **RTNORM** or **RTERROR**.

<b>sscut</b>	Selection set of objects to cut.
<b>boundary</b>	Object name of a single object or a selection set of multiple objects. Valid objects: line, polyline, circle, arc.
<b>skiptopo</b>	Skip flag: <b>1</b> Skip objects referenced by a topology <b>0</b> Trim objects referenced by a topology
<b>keepod</b>	Keep flag that sets whether to keep object data of clipped objects in result object: <b>1</b> Retain all object data on any clipped object <b>0</b> Drop object data on any clipped object
<b>ssreturn</b>	A selection set of cut objects.

This function cuts linear objects, such as lines, polylines, circles, and arcs, that cross the selected boundary. Unlike the [map\\_dwgtrimobj](#) function, this function does not delete the parts of the object on either side of the boundary. For example, you could mark a boundary and divide one map into two section maps along this boundary.

The following sample creates a filtered selection set of objects to be cut, (broken) and a filtered selection set containing one object to be used as the cutting edge. `Map_dwgBreakObj()` is called with all required parameters and if the operation is successful, the number of objects retained is reported. **Resbufs** are released as required and selection sets are freed.

```

struct resbuf* pFilteredCutObjsRb = acutBuildList(
    RTDXF0, "LWPOLYLINE",
    8, "Landbase",
    0);

ads_name ssObjsToCut;
acedSSGet("_X", NULL, NULL, pFilteredCutObjsRb, ssObjsToCut);

struct resbuf* pFilteredObjsToCutRb = acutBuildList(
    RTDXF0, "LWPOLYLINE",
    8, "MapEdge",
    0);

ads_name boundaryObj;
acedSSGet("X", NULL, NULL, pFilteredObjsToCutRb, boundaryObj);

short skipTopoObjs = 1; // Skip
short keepObjectData = 1; // Retain
ads_name ssObjsAreCut;

int resultCode = map_dwgBreakObj(
    ssObjsToCut,
    boundaryObj,
    skipTopoObjs,
    keepObjectData,
    ssObjsAreCut);

if (RTNORM == resultCode){
    long ssObjsAreCutLength;
    acedSSLength(ssObjsAreCut, &ssObjsAreCutLength);
    acutPrintf(
        "\n%d objects have been retained."
        , ssObjsAreCutLength);
}
else {
    acutPrintf(
        "\nUnable to complete the boundary break.");
}
acutRelRb(pFilteredCutObjsRb);
acutRelRb(pFilteredObjsToCutRb);
resultCode = acedSSFree(boundaryObj);

```



## map\_dwgtrimobj

### [Map Boundary Functions](#)

---

Trims linear objects inside or outside of a specified boundary.

int

```
map_dwgtrimobj(  
    ads_name ssclip,  
    ads_name boundary,  
    int inorout,  
    int skiptopo,  
    int keepod,  
    int bitflag,  
    ads_name ssreturn);
```

Returns **RTNORM** or **RTERROR**.

<b>ssclip</b>	Selection set of objects to trim.
<b>boundary</b>	Entity name of a boundary object. Valid objects: a single circle or a single closed 2D polyline.
<b>inorout</b>	Trim flag: <b>1</b> Trim outside boundary <b>0</b> Trim inside boundary
<b>skiptopo</b>	Skip flag: <b>1</b> Skip objects referenced by a topology <b>0</b> Trim objects referenced by a topology
<b>keepod</b>	Keep flag that sets whether to keep object data of trimmed objects in result object: <b>1</b> Drop object data from all trimmed objects <b>0</b> Retain object data on all trimmed objects
<b>bitflag</b>	Bit flag that sets the way to handle objects that cannot be trimmed: <b>0</b> Delete these objects within or on trim boundary <b>1</b> Ignore these objects within or on trim boundary <b>2</b> Reference the insertion point of any of these objects within or on trim boundary
<b>ssreturn</b>	A selection set of trimmed objects.

The following example prompts you to select an object to trim and make choices about the trim operation. It includes error reporting.

```
ads_name sscut, boundary, result;
ads_point point;
int ret, nb, i, skiptopo, keepod, inorout, bitflag;
char kword[50];

ads_printf ("\nSelect objects to trim : ");
if (ads_ssget (NULL, NULL, NULL, NULL, sscut) != RTNORM)
    return;

if (ads_entsel ("\nSelect boundary object : ", boundary, point) != RTNORM) {
    ads_ssfree (sscut);
    return;
}

ads_initget (0, "Yes No");
ret = ads_getkword ("\nSkip objects referenced by a topology Yes/No <Yes> : ", kword);
if (ret != RTNORM && ret != RTNONE) {
    ads_ssfree (sscut);
    return;
}

if (ret == RTNONE || kword[0] == 'Y')
    skiptopo = 1;
else
    skiptopo = 0;

ads_initget (0, "Yes No");
ret = ads_getkword ("\nRetain object data Yes/No <Yes> : ", kword);
if (ret != RTNORM && ret != RTNONE) {
    ads_ssfree (sscut);
    return;
}

if (ret == RTNONE || kword[0] == 'Y')
    keepod = 0;
else
    keepod = 1;

ads_initget (RSG_NONULL, "Inside Outside");
```

```

ret = ads_getkword ("\nTrim Inside or Outside : ", kword);
if (ret != RTNORM) {
    ads_ssfree (sscut);
    return;
}

if (kword[0] == 'I')
    inorout = 0;
else
    inorout = 1;

ads_initget (0, "Delete Ignore insertionPoint");
ret = ads_getkword ("\nObjects that cannot be trimmed Delete/Ignore/insertionPoint : ", kword);
if (ret != RTNORM && ret != RTNONE) {
    ads_ssfree (sscut);
    return;
}

if (ret == RTNONE || kword[0] == 'P')
    bitflag = 2;
else if (kword[0] == 'I')
    bitflag = 1;
else
    bitflag = 0;

ret = map_dwgtrimobj (sscut, boundary, inorout, skiptopo, keepod, bitflag, result);
if (ret == RTNORM) {
    ads_printf ("\nObject(s) successfully trimmed.");
    ads_ssfree (result);
}
else {
    nb = ade_errqty ();
    for (i = 0; i < nb; i++)
        ads_printf ("\nError %d of %d : %s", i + 1, nb, ade_errmsg (i));
}
ads_ssfree (sscut);

```



## map\_pltblkatts

### [Plotting Functions](#)

---

Gets a list of block attributes.

```
int
map_pltblkatts(
    char *name,
    struct resbuf **res);
```

Returns **RTNORM** or **RTERROR**.

**name**      Layout block name.

**res**        Pointer to pointer to result buffer of block attributes.

If the function returns **RTNORM**, you must release the **resbuf**.

The following sample populates a **resbuf** with plot layout block attribute name(s) using **map\_pltBlkAtts()**. If the operation is successful the attribute name(s) are displayed, and the **resbuf** is released as required.

```
char* pszLayoutBlockName = "TITLE-A";
struct resbuf* pLayoutBlockAttsRb = NULL;
int returnCode = map_pltBlkAtts(pszLayoutBlockName, &pLayoutBlockAttsRb);
if (RTNORM == returnCode){
    struct resbuf* rb = pLayoutBlockAttsRb;
    while(NULL != rb) {
        acutPrintf(
            "\nThe following block attribute has been found on the layout block \"%s\": %s"
            , pszLayoutBlockName, rb->resval.rstring);
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(
        "\nNo block attributes have been found on the layout block \"%s\". ");
}
acutRelRb(pLayoutBlockAttsRb);
```





## map\_pltblklist

### [Plotting Functions](#)

---

Gets a list of valid plot layouts for the current work session.

```
struct resbuf
*map_pltblklist(
    );
```

Returns a result buffer of block names or **NULL**

You must release the **resbuf**.

This function returns the plot layout (block) names that are usable as plot layouts. To qualify, a block must have at least one unique viewport on one of its layers. That is, if the block has more than one viewport, it must have one layer that contains only one viewport.

The unique viewport can share its layer with objects of other types, such as lines, polylines, blocks, and text.

The following sample populates a **resbuf** with the plot layout block name using **map\_pltBlkList()**. If the operation is successful the layout block name is displayed, and the **resbuf** is released as required.

```
struct resbuf* pLayoutBlockNameRb = NULL;
pLayoutBlockNameRb = map_pltBlkList();
if(pLayoutBlockNameRb != NULL) {
    acutPrintf(
        "\nThe plot layout block is named: %s."
        , pLayoutBlockNameRb->resval.rstring);
}
else {
    acutPrintf(
        "\nNo plot layout blocks are defined.");
}
acutRelRb(pLayoutBlockNameRb);
```



## map\_pltblkvps

### [Plotting Functions](#)

---

Returns a list of valid viewport layers in a specified layout blocks.

```
int
map_pltblkvps(
    char *name,
    struct resbuf **res);
```

Returns **RTNORM** if the list is initialized or **RTERROR** if the block name or layout block is invalid.

**name**      Layout block name.  
**res**        Pointer to pointer to result buffer of viewport layers.

If the function returns **RTNORM**, you must release the **resbuf**.

The following sample populates a **resbuf** with plot layout block viewport layer name(s) using **map\_pltBlkVps()**. If the operation is successful the viewport layer name(s) are displayed, and the **resbuf** is released as required.

```
char* pszLayoutBlockName = "TITLE-A";
struct resbuf* pLayoutBlockVportLayerRb = NULL;
int returnCode = map_pltBlkVps(pszLayoutBlockName, &pLayoutBlockVportLayerRb);
if (RTNORM == returnCode){
    struct resbuf* rb = pLayoutBlockVportLayerRb;
    while(NULL != rb) {
        acutPrintf(
            "\nThe following viewport layer(s) have been found on the layout block \"%s\": %s"
            , pszLayoutBlockName, rb->resval.rstring);
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(
        "\nNo viewport layers have been found on the layout block \"%s\". ");
}
```

```
acutRelRb(pLayoutBlockVportLayerRb);
```



## map\_pltcleanup

[Plotting Functions](#)

---

Restores settings altered by `map_pltinit`.

int

```
map_pltcleanup(  
    );
```

Returns `RTNORM` or `RTERROR`.

This function restores certain settings to the state they were in before `map_pltinit` was called. See [map\\_pltInit](#) for a list of affected settings. Before you can use other plot functions, you must call `map_pltinit` again.



## map\_pltcurrdef

[Plotting Functions](#)

---

Selects or creates a plot set.

```
int  
map_pltcurrdef(  
    char *name);
```

Returns **RTNORM** or **RTERROR**.

**name**     The name of the plot set.

If a plot set called **name** does not exist in the current work session or if the function [map\\_pltdefread](#) was not called, this function creates a new plot set called **name**.

Use [map\\_pltcurrset](#) to define the attributes of the plot set.

If the plot set **name** exists and the function [map\\_pltdefread](#) was called, this function loads a copy of **name** into memory. Use [map\\_pltcurrGet](#) and [map\\_pltcurrSet](#) to examine or change its attributes.

**Note** Only one plot set can be current.



## map\_pltcurrdel

[Plotting Functions](#)

---

Resets a plot set attribute to its default value.

int

```
map_pltcurrdel(  
    char *attr);
```

Returns **RTNORM** or **RTERROR**.

**attr**     The name of the attribute to reset.



## map\_pltcurrget

### [Plotting Functions](#)

---

Gets an attribute value for the current plot set.

```
int
map_pltcurrget(
    char *attr,
    struct resbuf **res);
```

Returns **RTNORM** or **RTERROR**.

**attr**     The name of the [plot set attribute](#) to retrieve.

**res**       **Resbuf** containing the value of the attribute.

This function retrieves the value of a specific attribute for the current plot set. The data type of the return value depends upon the attribute. To set plot set attributes, use [map\\_pltdefget](#).

If the function returns **RTNORM**, you must release the **resbuf**.

The following sample populates a **resbuf** with the plot query name using [map\\_pltCurrGet\(\)](#). If the operation is successful the query name is displayed, and the **resbuf** is released as required.

```
char* pszPlotSetAttrb = "name";
struct resbuf* pPlotSetAttrbValueRb = NULL;
int returnCode = map_pltCurrGet(pszPlotSetAttrb, &pPlotSetAttrbValueRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe plot set attribute \"%s\" contained the following value: \n\t\"%s\"""
        , pszPlotSetAttrb, pPlotSetAttrbValueRb->resval.rstring);
}
else {
    acutPrintf(
        "\nThe plot set attribute \"%s\" contained no value."
        , pszPlotSetAttrb);
}
acutRelRb(pPlotSetAttrbValueRb);
```



## map\_pltcurrsave

[Plotting Functions](#)

---

Appends the current plot set definition to the plot set list.

```
int  
map_pltcurrsave();
```

Returns **RTNORM** or **RTERROR**.

This function appends the current plot set definition to the plot set list of the current work session. This list is not saved in the work session until the user executes a save with a call to [map\\_pltdefsave](#).

### To edit and save a plot set definition

1. Get the plot set definition with [map\\_pltdefread](#) and [map\\_pltcurrdef](#).
2. Make the necessary changes.
3. Save the definition in the list with **map\_pltcurrsave**.
4. Save the list in the current work session with [map\\_pltdefsave](#).
5. Save the work session.

If you do not save before the end of the processing, another application can overwrite your changes with a call to [map\\_pltcurrdef](#).



## map\_pltcurrset

### [Plotting Functions](#)

---

Sets the value of an attribute for the current plot set.

```
int
map_pltcurrset(
    char attr,
    struct resbuf *res);
```

Returns **RTNORM** or **RTERROR**.

**attr**     The name of the plot attribute to set.  
**res**       Pointer to **resbuf** for attribute value.

If the function returns **RTNORM**, you must release the **resbuf**.

The following sample creates a **resbuf** containing a value representing the plot set description.

**Map\_pltCurrSet()** is called with all required parameters, if the operation is successful the plot set attribute name and the updated value are displayed. The **resbuf** is then released as required.

```
char* pszPlotSetAttrb = "desc";
struct resbuf* pPlotSetAttrbValueRb = acutBuildList(
    RTSTR,
    "A new plot set description",
    0);
int returnCode = map_pltCurrSet(
    pszPlotSetAttrb,
    pPlotSetAttrbValueRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\n\nThe plot set attribute \"%s\" has been set to the following value: \n\t\"%s\""\n",
        pszPlotSetAttrb, pPlotSetAttrbValueRb->resval.rstring);
}
else {
    acutPrintf(
        "\n\nThe plot set attribute \"%s\" was not set."

```

```
    , pszPlotSetAttrb);  
}  
acutRelRb(pPlotSetAttrbValueRb);
```



## map\_pltdefdelete

[Plotting Functions](#)

---

Deletes a plot set definition.

int

```
map_pltdefdelete(  
    char *name);
```

Returns **RTNORM** or **RTERROR**.

**name**     The name of the plot set.

The function updates the plot definition dictionary in the current work session.



## map\_pltdefget

### [Plotting Functions](#)

---

Gets the value of an attribute of the plot set definition.

```
int
map_pltdefget(
    char *name,
    char *attr,
    struct resbuf **res);
```

Returns **RTNORM** or **RTERROR**.

**name**      The name of the plot set.

**attr**      Name of the [plot set attribute](#) to retrieve.

**res**        Pointer to a pointer to the **resbuf** containing the given [plot set attribute](#)

This function returns the value of the attribute through the **resbuf** parameter. If the function returns **RTNORM**, you must release the **resbuf**.

To get the value of an attribute for the current plot set, use [map\\_pltCurrGet](#).

The data type of the return value depends on the attribute.

The following sample populates a **resbuf** with the value of the plot set description for a specified plot set. **Map\_pltDefGet()** is called with all required parameters, and if the operation is successful the specified plot set and description are displayed. The **resbuf** is then released as required.

```
char* pszPlotSetName = "ADSRX_Sample";
char* pszPlotSetAttrb = "desc";
struct resbuf* pPlotSetAttrbValueRb = NULL;
int returnCode = map_pltDefGet(pszPlotSetName, pszPlotSetAttrb, &pPlotSetAttrbValueRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\n\nThe \"%s\" plot set attribute; \"%s\" contains the following value: \n\t\"%s\""\n\n",
        pszPlotSetName, pszPlotSetAttrb, pPlotSetAttrbValueRb->resval.rstring);
}
```

```
else {
    acutPrintf(
        "\nThe plot set attribute \"%s\" could not be obtained."
        , pszPlotSetAttrb);
}
acutRelRb(pPlotSetAttrbValueRb);
```



## map\_pltdeflist

### [Plotting Functions](#)

---

Gets the names of available plot set definitions.

```
int
map_pltdeflist(
    struct resbuf **res);
```

Returns **RTNORM** or **RTERROR**.

**res**      Pointer to pointer to **resbuf** for list of returned plot set definitions.

This function returns the value of the attribute through the **resbuf** parameter. If the function returns **RTNORM**, you must release the **resbuf**.

The following sample populates a **resbuf** with the plot set names using **map\_pltDefList()**. If the operation is successful the plot set name(s) are displayed, and the **resbuf** is released as required.

```
struct resbuf* pPlotSetListRb = NULL;
int returnCode = map_pltDefList(&pPlotSetListRb);
if (RTNORM == returnCode){
    struct resbuf* rb = pPlotSetListRb;
    acutPrintf(
        "\n\nThe current project contains the following plot set(s): ");
    while(NULL != rb) {
        acutPrintf(
            "\n\n\"%s\"", rb->resval.rstring);
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(
        "\n\nThe current project contains no plot sets.");
}
acutRelRb(pPlotSetListRb)
```



## map\_pltdefread

[Plotting Functions](#)

---

Reads in a plot set definition.

int

```
map_pltdefread();
```

Returns **RTNORM** or **RTERROR**.

This function provides access to plot sets in the plot definition dictionary for the current work session.

**Warning** If you do not call [map\\_pltdefsave](#), a new call to **map\_pltdefread** will erase your new plot set definition or your changes to an existing plot set.



## map\_pltdefsave

[Plotting Functions](#)

---

Writes the current plot set definition to the plot definition dictionary.

```
int  
map_pltdefsave(  
    );
```

Returns **RTNORM** or **RTERROR**.

This function stores the plot set definition list in the current work session.

If you do not call this function, a new call to [map\\_pltdefread](#) will erase your new plot set definition or your changes to an existing plot set.



## map\_pltdefvalid

[Plotting Functions](#)

---

Performs a cursory check of the validity of a plot set.

int

```
map_pltdefvalid(  
    char *name);
```

Returns **RTNORM** or **RTERROR**.

**name**     The name of the plot set.

The function performs a cursory check of the given plot set to see if all required attributes have been set. It does not attach and query the boundary drawing. When used within a dialog box, this call can quickly check on a plot set's usability.

If the check fails, you can use data extension error message functions ([ade\\_err\[xx\]](#)) to retrieve errors from the error stack.

To perform an extensive check, use the [map\\_pltdefverify](#) function.



## map\_pltdefverify

[Plotting Functions](#)

---

Performs an extensive check of the validity of a plot set.

```
int  
map_pltdefverify(  
    char *name);
```

Returns **RTNORM** or **RTERROR**.

**name**     The name of the plot set.

This function makes sure that no errors occur when plotting takes place. Since the check includes querying for all boundary objects, it could take some time.

To perform a cursory check, use the [map\\_pltdefvalid](#) function.

If the check fails, you can use data extension error message functions ([ade\\_errfxx](#)) to retrieve errors from the error stack.



## map\_pltdisplay

[Plotting Functions](#)

---

Generates the plot display for the specified boundary.

int

```
map_pltdisplay(  
    char *bndryname);
```

Returns **RTNORM** or **RTERROR**.

**bndryname**      The name of a boundary object.

This function prepares the display to plot for the given plot set definition and boundary object name. Generating the plot display for the specified boundary includes

- Switching to paper mode (if necessary)
- Inserting the layout block
- Mapping boundary object data to layout block attributes (if applicable)
- Executing the query(ies) to collect the objects to plot
- Trimming the objects to the boundary (if applicable)
- Displaying the objects in the view port

You must call **map\_pltdisplay** before you call [map\\_pltplot](#).



## map\_pltexecute

[Plotting Functions](#)

---

Plots the plot set for the specified plot set name.

int

```
map_pltexecute(  
    char *name);
```

Returns **RTNORM** or **RTERROR**.

**name**     The name of the plot set to execute.

This function generates and issues plots for each defined boundary.

To get a list of available plot set names, use [map\\_pltdeflist](#).



## map\_pltinit

### [Plotting Functions](#)

---

Initializes environment for plotting.

```
int  
map_pltinit();
```

Returns **RTNORM** always.

This function must be called before any other plotting functions. The **map\_pltinit** function modifies the following settings and checks that the plot set description file exists:

AutoCAD Variable	Setting
CMDECHO	0
expert	1

Data Extension Preference	Setting
DontAddObjectsToSaveSet	T
ActivateDwgsOnAttach	T
MkSelSetWithQryObj	T

To restore these settings to their original values, use [map\\_pltcleanup](#). Before you can use other plot functions, you must call **map\_pltinit** again.



## map\_pltplot

[Plotting Functions](#)

---

Executes the current plot set script.

int

```
map_pltplot();
```

Returns **RTNORM** or **RTERROR**.



## map\_pltrestore

[Plotting Functions](#)

---

Restores display altered by `map_pltdisplay`.

```
int  
map_pltrestore();
```

Returns `RTNORM` or `RTERROR`.



## map\_topoaudit

### [Map Topology Functions](#)

---

Determines whether the specified topology is correct.

```
int  
map_topoaudit(  
    ade_id tpm_id);
```

Returns **RTNORM** or **RTERROR**.

**tpm\_id** The ID of the topology to check for correctness. The topology must be open for Read.

The function audits the geometry of a topology to determine whether the geometrical relationships defined by the topology object data are correct. It shows the location of errors.

The following sample checks a topology named "I-95\_Buffer" and provides error reporting based on the results of the check.

```
char* pszTopoName = "I-95_Buffer";  
int resultCode = tpm_acload(pszTopoName, 0);  
ade_id topoId = tpm_acopen(  
    pszTopoName,  
    0);  
if (ADE_NULLID != topoId)  
{  
    resultCode = ade_errclear();  
    int resultCode = map_topoAudit(topoId);  
    if (RTNORM == resultCode) {  
        acutPrintf(  
            "\nTopology %s is correct and complete."  
            , pszTopoName);  
    }  
    else {  
        int errorStkQty = ade_errqty();  
        if (errorStkQty > 0)  
        {
```

```
    for (int i = 0; i < errorStkQty; i++)
    {
        int nErrCode = ade_errcode(i);
        char* pszErrorMsg = ade_errmsg(i);
        acutPrintf ("\n%s", pszErrorMsg);
    }
}
}
else {
    acutPrintf(
        "\nThe topology was not opened.");
}
resultCode = tpm_acclose(topoId);
resultCode = tpm_acunload(pszTopoName);
```



## map\_topoclose

### [Map Topology Functions](#)

---

Converts all polygons in a topology to closed polylines.

```
int
map_topoclose(
    char *topoName,
    char *layer,
    int group,
    int odata,
    int aselink,
    ads_name ssreturn);
```

Returns **RTNORM** or **RTERROR**.

<b>topoName</b>	The name of a topology. The topology must be closed.
<b>layer</b>	The name of the layer that will contain the closed polylines. If the layer name is incorrect (for example "@@"), polylines are created on the current layer.
<b>group</b>	Group flag for complex polygons: <b>1</b> or <b>0</b> . <b>1</b> group complex polygons <b>0</b> do not group complex polygons
<b>odata</b>	Object data flag that sets whether to copy object data on the polygon centroid to the resulting polyline: <b>1</b> or <b>0</b> . <b>1</b> copy object data on the centroid <b>0</b> ignore object data on the centroid
<b>aselink</b>	ASE link flag that sets whether to copy ASE link data on the centroid to the resulting polyline: <b>1</b> or <b>0</b> . <b>1</b> copy ASE link data on the centroid <b>0</b> ignore ASE link data on the centroid
<b>ssreturn</b>	The returned selection set containing the closed polylines.

This function passes the selection set through the *ssreturn* parameter. If the function returns **RTNORM**, you must release the returned selection set with the **ads\_ssfree** function.



## map\_topocomplete

### [Map Topology Functions](#)

---

Completes all objects in a loaded partial topology.

```
int  
map_topocomplete(  
    char *toponame,  
    ads_name ssreturn);
```

Returns **RTNORM** or **RTERROR**.

**toponame**     The name of the topology to complete. The topology must be closed.  
**ssreturn**     The returned selection set of objects retrieved to complete the topology.

This function returns the selection set through the *ssreturn* parameter. If the function returns **RTNORM**, you must release the returned selection set with the *ads\_ssfree* function.

This function performs a query to retrieve objects into the work session. The [tpm\\_infocomplete](#) function determines whether the topology is completely represented in the work session.

This function can only complete objects imported from an existing source drawing. For a polygon topology, this function imports links, nodes, and a centroid, if it is missing, from the source drawing. For a network topology, this function imports links and nodes. For a network topology, it imports only nodes.



## map\_topostat

### [Map Topology Functions](#)

---

Gets statistics about a topology.

```
struct resbuf
*map_topostat(
    ade_id tpm_id);
```

Returns a **resbuf** list containing the statistics for the specified topology, or **NULL** if an error occurs.

**tpm\_id**     The ID of the topology for which to get statistics. The topology must be open for read.

You must release the **resbuf**.

The following tables show statistic names, the data type and the topology types they apply to:

#### All Topologies

node_count	RTSHORT
link_count	RTSHORT
polygon_count	RTSHORT
min_x	RTREAL
min_y	RTREAL
max_x	RTREAL
max_y	RTREAL

**Note** This function is not designed to count polygons in a partial topology. If the topology in question is partial, the **polygon\_count** statistic may be overstated. This is because **map\_topostat** counts not only the polygons in the partial topology, but also any polygons that share common edges with them in the complete topology, even if the adjacent polygons are not actually present in the current drawing.

#### Network Topologies

---

length_total	RTREAL
length_average	RTREAL
length_min	RTREAL
length_max	RTREAL
length_variance	RTREAL
length_deviation	RTREAL

### Polygon Topologies

area_total	RTREAL
area_average	RTREAL
area_min	RTREAL
area_max	RTREAL
area_variance	RTREAL
area_deviation	RTREAL
perimeter_total	RTREAL
perimeter_average	RTREAL
perimeter_min	RTREAL
perimeter_max	RTREAL
perimeter_variance	RTREAL
perimeter_deviation	RTREAL

The following sample loads and then opens a network topology for read using `tpm_acload()` and `tpm_acopen()` respectively. A `resbuf` is populated with that topologies information, then is displayed. The `resbuf` is released as required.

```
char* pszTopoName = "NetTopo";
int topoWriteAccess = 0;
int returnCode = tpm_acload(pszTopoName, 0);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
struct resbuf* pTopoStatsRb = map_topoStat(topoId);
if (NULL != pTopoStatsRb){
```

```

struct resbuf* rb = pTopoStatsRb;
while(NULL != rb) {
    if (rb->restype == RTSTR) {
        acutPrintf(
            "\nThe \"%s\" property contained the value:"
            , rb->resval.rstring);
        if (NULL != (rb = rb->rbnext)){
            if (rb->restype == RTDOTE) {
                rb = rb->rbnext;
                switch(rb->restype)
                {
                    case RTSTR:
                        acutPrintf(
                            "\"%s\" RTSTR"
                            , rb->resval.rstring);
                        break;
                    case RTREAL:
                        acutPrintf(
                            " %.2lf RTREAL"
                            , rb->resval.rreal);
                        break;
                    case RTSHORT:
                        acutPrintf(
                            " %d RTSHORT"
                            , rb->resval.rint);
                        break;
                    case RTLONG:
                        acutPrintf(
                            " %d RTLONG"
                            , rb->resval.rlong);
                        break;
                    default:
                        break;
                }
            }
        }
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(

```

```
        "\nNo topologies are accessible in this project.");  
    }  
    acutRelRb(pTopoStatsRb);  
    tpm_acclose(topoId);  
    tpm_acunload(pszTopoName);
```



## Boundary Functions

[Map Book Function Synopsis](#)

---

The map boundary functions begin with `map_dwg`.

<a href="#">map_dwgbreakobj</a>	Breaks objects where they cross boundary edges.
<a href="#">map_dwgtrimobj</a>	Trims linear objects inside or outside of a specified boundary.



## Plotting Functions

### [Map Book Function Synopsis](#)

---

The map plotting functions begin with `map_plt`.

<a href="#">map_pltblkatts</a>	Gets a list of block attributes.
<a href="#">map_pltblklist</a>	Returns a sorted list of block names that are usable as plot layouts.
<a href="#">map_pltblkvps</a>	Returns a list of valid viewport layers in layout blocks.
<a href="#">map_pltcleanup</a>	Restores settings altered by <code>map_pltinit</code> .
<a href="#">map_pltcurrdef</a>	Selects or creates a plot set.
<a href="#">map_pltcurrdel</a>	Resets a plot set attribute to its default value.
<a href="#">map_pltcurrget</a>	Retrieves the value for a specific attribute for the current plot set.
<a href="#">map_pltcursave</a>	Appends the current plot set definition to the plot set list.
<a href="#">map_pltcurrset</a>	Sets the value of an attribute for the current plot set.
<a href="#">map_pltdefdelete</a>	Deletes a plot set definition.
<a href="#">map_pltdefget</a>	Gets the value of an attribute of the plot set definition.
<a href="#">map_pltdeflist</a>	Returns a list of available plot set definitions in the project.
<a href="#">map_pltdefread</a>	Reads in a plot set definition from the plot definition dictionary for the project.
<a href="#">map_pltdefsave</a>	Writes the current plot set definition to the plot definition dictionary.
<a href="#">map_pltdefvalid</a>	Tests the plot set definition for validity.
<a href="#">map_pltdefverify</a>	Validates the given plot set to prevent plotting errors.
<a href="#">map_pltdisplay</a>	Generates the plot display for the specified boundary.
<a href="#">map_pltexecute</a>	Executes a plot, given a specified plot set.
<a href="#">map_pltinit</a>	Initializes environment for plotting.

<a href="#">map_pltplot</a>	Executes the plot script of the current plot set definition.
<a href="#">map_pltrestore</a>	Restores display altered by <a href="#">map_pltdisplay</a> .



## Topology Functions

[Map Book Function Synopsis](#)

---

The map topology functions begin with `map_topo`.

<a href="#">map_topoaudit</a>	Checks the geometrical relationships defined by the topology object data.
<a href="#">map_topoclose</a>	Converts all polygons in a topology to closed polylines.
<a href="#">map_topocomplete</a>	Completes all objects in a loaded partial topology.
<a href="#">map_topostat</a>	Gets the statistics for a topology.

Links

[AcDbBasicFilter Class](#)

AcDbBasicFilter:: ELayerStatus Enumeration

[AcDbBasicFilter Class](#)

Enumerates the types of layer-status filters.

```
enum ELayerStatus {  
    eIncludeFrozenLayers = 0x0001,  
    eIncludeLockedLayers = 0x0002,  
    eIncludeOffLayers = 0x0004
```

```
};
```

File

AcDbObjectFilter.h

Parameters	Description
eIncludeFrozenLayers	Filter objects on layers that are frozen.
eIncludeLockedLayers	Filter objects on layers that are locked.
eIncludeOffLayers	Filter objects on layers that are turned off.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)  
AcDbBasicFilter:: ~AcDbBasicFilter Destructor  
[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Destroys an instance of this class.

```
virtual ~AcDbBasicFilter();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter::AcDbBasicFilter Constructor](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Constructs an instance of this class with the default value filter values set to "\*" to filter all entities.

```
AcDbBasicFilter();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter:: Blocks Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Retrieves the block(s) on which to filter. The block filter is defined with SetBlocks() (twoforms).

```
const ACHAR* Blocks() const;
```

Returns

Returns the blocks on which to filter as any of the following values: A single block name ("Block1", for example)A list of comma-separated block names ("Block1, Block2", for example)A "\*" indicating all blocksA "" (empty string) indicating that blocks are ignored during filteringBlock names with wildcard characters ("Block?" or "Block-\*", for example)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter::FeatureClasses Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Retrieves the feature class(es) on which to filter. The feature-class filter is defined with SetFeatureClasses() (twoforms).

```
const ACHAR* FeatureClasses() const;
```

Returns

Returns the feature classes on which to filter as any of the following values: A single feature-class name ("Feature1", for example)A list of comma-separated feature-class names ("Feature1,Feature2", for example)A "\*" indicating all feature classesA "" (empty string) indicating that feature classes are ignored during filteringFeature-class names with wildcard characters ("Feature?" or "Feature-\*", for example)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter::FilterObjects Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Filters objects on the current drawing based on the layer, feature-class, and block criteria.

A filtering criterion can take the following values: "\*" - (Default) Filters all entities that reside on layers with normal layer status. Unless SetLayerStatusMask() has been called with specific layer mask settings, all objects on layers which are not frozen, off, or locked are considered in the filtering process. "" (empty string) - Indicates that a particular filter is to be ignored during the filtering process. NULL - Indicates that particular filtered entities are to be excluded from the filtered set. A name - The name of a single entity. "Layer1", for example. Multiple names - A list of comma-separated entity names. "Block1,Block2", for example. Wildcards - "?" matches any single character and "\*" matches zero or more characters. "Feature?" and "Feature-\*", for example. .

```
virtual Acad::ErrorStatus FilterObjects(  
    AcDbObjectIdArray& outputIds,  
    const AcDbObjectIdArray& inputIds  
) const;
```

Parameters	Description
outputIds	Output array of the IDs of the objects that met the criteria and were filtered. You are responsible for checking the length of this array; a valid filter combination can return no filtered objects.
inputIds	Input array of the IDs of the candidate objects to inspect for matching filter criteria.

## Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if the filter fails.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter::Layers Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Retrieves the layer(s) on which to filter. The layer filter is defined with SetLayers() (twoforms). NULL and empty strings ("") are invalid values.

```
const ACHAR* Layers() const;
```

Returns

Returns the layers on which to filter as any of the following values: A single layer name ("Layer1", for example)A list of comma-separated layer names ("Layer1,Layer2", for example)A "\*" indicating all layers except off, locked, or frozen onesLayer names with wildcard characters ("layer?" or "lay\*", for example)

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: LayerStatusMask Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Determines whether the layer status mask has been set by using `SetLayerStatusMask()`. A nonzero return value indicates that objects on layers which are frozen, locked, or off are to be considered in the filtering process. A zero return value indicates that no layer mask is set and filtering will be applied to objects with normal layer status.

```
int LayerStatusMask() const;
```

Returns

Returns nonzero (true) if the layer mask is set, or zero (false) if it is not set.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

AcDbBasicFilter:: ResetBlocks Method

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Clears the filter list of blocks.

```
Acad::ErrorStatus ResetBlocks();
```

Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Remarks

This function sets the block filter to the default value "\*" (filter all blocks).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)  
AcDbBasicFilter:: ResetFeatureClasses Method  
[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Clears the filter list of feature classes.

```
Acad::ErrorStatus ResetFeatureClasses();  
Returns
```

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

## Remarks

This function sets the feature-class filter to the default value "\*" (filter all feature classes).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: ResetLayers Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Clears the list of filter layers.

```
Acad::ErrorStatus ResetLayers();
```

Returns

Returns Acad::ErrorStatus eOk if successful. Returns Acad::ErrorStatus eInvalidInput if parameter validation failed.

Remarks

This function sets the layer filter to the default value "\*" (filter all layers).

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbBasicFilter Class](#), [AcDbBasicFilter Class](#)

[AcDbBasicFilter:: SetLayerStatusMask Method](#)

[AcDbBasicFilter Class](#) | [AcDbBasicFilter Class](#)

Sets the layer-status filter criterion that specifies whether objects on frozen, locked, or off layers are filtered. Use `LayerStatusMask()` to determine whether the layer-status filter is set.

```
Acad::ErrorStatus SetLayerStatusMask(  
    const int statusMask  
);
```

Parameters	Description
statusMask	Input ELayerStatusmask bits of the criteria to set.

Returns

Returns `Acad::ErrorStatus eOk` if successful. Returns `Acad::ErrorStatus eInvalidInput` if the input mask contains invalid `ELayerStatus` enum values.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilter Class](#), [AcDbObjectFilter Class](#)

[AcDbObjectFilter:: ~AcDbObjectFilter Destructor](#)

[AcDbObjectFilter Class](#) | [AcDbObjectFilter Class](#)

Destroys an instance of this class.

```
virtual ~AcDbObjectFilter();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbObjectFilter Class](#), [AcDbObjectFilter Class](#)

[AcDbObjectFilter::AcDbObjectFilter Constructor](#)

[AcDbObjectFilter Class](#) | [AcDbObjectFilter Class](#)

Constructs an instance of this class.

```
AcDbObjectFilter();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilter Class](#), [AcDbObjectFilter Class](#)

[AcDbObjectFilter::FilterObjects Method](#)

[AcDbObjectFilter Class](#) | [AcDbObjectFilter Class](#)

Filters objects on the current drawing based on the filtering criteria.

```
virtual Acad::ErrorStatus FilterObjects(  
    AcDbObjectIdArray& outputIds,  
    const AcDbObjectIdArray& inputIds  
) const = 0;
```

Parameters	Description
outputIds	Output array of the IDs of the objects that met the criteria and were filtered.
inputIds	Input array of the IDs of the candidate objects to inspect for matching filter criteria.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilter Class](#), [AcDbObjectFilter Class](#)

AcDbObjectFilter::IsActive Method

[AcDbObjectFilter Class](#) | [AcDbObjectFilter Class](#)

Determines whether this filter is active.

```
bool IsActive() const;
```

Returns

Returns true if this filter is active, or false if it is inactive.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbObjectFilter Class](#), [AcDbObjectFilter Class](#)

AcDbObjectFilter::SetActive Method

[AcDbObjectFilter Class](#) | [AcDbObjectFilter Class](#)

Activates or deactivates this filter. When this filter is active, objects are filtered based on the filtering criteria when FilterObjects() in the derived class is called. When this filter is inactive, no objects are filtered. This setting applies only when the [AcDbObjectFilterGroup](#) class is used to filter objects.

```
Acad::ErrorStatus SetActive(  
    bool bActive  
);
```

Parameters	Description
bActive	Input true to activate this filter, or false to deactivate it.

## Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)  
AcDbObjectFilterGroup:: ~AcDbObjectFilterGroup Destructor  
[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Destroys an instance of this class.

```
virtual ~AcDbObjectFilterGroup();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)  
AcDbObjectFilterGroup:: AcDbObjectFilterGroup Constructor  
[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Constructs an instance of this class.

```
AcDbObjectFilterGroup();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

[AcDbObjectFilterGroup:: AddObjectFilter Method](#)

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Adds a filter to the end of the filter list.

```
Acad::ErrorStatus AddObjectFilter(  
    const AcDbObjectFilter& objFilter  
);
```

Parameters	Description
objFilter	Input AcDbObjectFilter to add.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

AcDbObjectFilterGroup::FilterObjects Method

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Filters objects on the current drawing based on the filtering criteria of all the filters in the filter list.

```
virtual Acad::ErrorStatus FilterObjects(  
    AcDbObjectIdArray& outputIds,  
    const AcDbObjectIdArray& inputIds  
) const;
```

Parameters	Description
outputIds	Output array of the IDs of the objects that met the criteria and were filtered.
inputIds	Input array of the IDs of the candidate objects to inspect for matching filter criteria.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

AcDbObjectFilterGroup::GetObjectFilter Method

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Retrieves a filter from the filter list.

```
Acad::ErrorStatus GetObjectFilter(  
    const AcDbObjectFilter*& objFilter,  
    int index  
) const;
```

Parameters	Description
objFilter	Output retrieved AcDbObjectFilter.
index	Input list index of the filter to retrieve.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns Acad::ErrorStatus eOutOfRange.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

AcDbObjectFilterGroup::InsertObjectFilter Method

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Inserts a filter into the filter list.

```
Acad::ErrorStatus InsertObjectFilter(  
    int index,  
    const AcDbObjectFilter& objFilter  
);
```

Parameters	Description
index	Input list index at which to insert the filter.
objFilter	Input AcDbObjectFilter to insert.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns Acad::ErrorStatus eOutOfRange.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

AcDbObjectFilterGroup::IsEmpty Method

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Determines whether the filter list is empty.

```
bool IsEmpty() const;
```

Returns

Returns true if the filter list is empty, or false if it is not empty.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

[AcDbObjectFilterGroup:: ObjectFilterCount Method](#)

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Counts the number of filters in the filter list.

```
int ObjectFilterCount() const;
```

Returns

Returns the number of filters in the list.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

AcDbObjectFilterGroup:: RemoveAllObjectFilter Method

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Clears the filter list.

```
Acad::ErrorStatus RemoveAllObjectFilter();
```

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcDbObjectFilterGroup Class](#), [AcDbObjectFilterGroup Class](#)

[AcDbObjectFilterGroup:: RemoveObjectFilter Method](#)

[AcDbObjectFilterGroup Class](#) | [AcDbObjectFilterGroup Class](#)

Removes a filter from the filter list.

```
Acad::ErrorStatus RemoveObjectFilter(  
    int index  
);
```

Parameters	Description
index	Input list index of the filter to remove.

Returns

Returns Acad::ErrorStatus eOk if successful; otherwise, returns Acad::ErrorStatus eOutOfRange.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## AcMapOSEGetConnection global function

Gets the connection object.

Include AdMapOracleConnection.h.

```
AcMapOSEConnection *  
AcMapOSEGetConnection();
```

Returns the connection object.

Before you can do anything with Oracle Spatial data, you must get the AcMapOSEConnection object.

```
AcMapOSEConnection *pConnection = AcMapOSEGetConnection();
```

See [Managing the connection \(overview\)](#).

# AcMapOSEConnection class

[Functions](#) [Related Classes](#) [See Also](#)

Manages the Oracle database connection and associated reactors.

Instantiated automatically.

Accessed by AcMapOSEGetConnection(), a global function.

Include AdMapOracleConnection.h.

## Overviews

[Managing the connection](#) [Managing reactors](#)

## Code samples

[Connecting to an Oracle Database](#)

## Functions

[~AcMapOSEConnection](#)

[AddConnectionReactor](#)

[AddExportReactor](#)

[AddImportReactor](#)

[Connect](#)

[Database](#)

[Disconnect](#)

[Features](#)

[IsConnected](#)

[IsSchemaValid](#)

[RefreshSchema](#)

[RemoveConnectionReactor](#)

[RemoveExportReactor](#)

[RemoveImportReactor](#)

[Schema](#)

[Service](#)

[UserName](#)

# AcMapOSEConnectionReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom connection reactors.

Include AdMapOracleReactor.h.

To monitor and react to connection events, subclass a custom reactor from AcMapOSEConnectionReactor, and then add an instance of it to your application using AcMapOSEConnection::AddConnectionReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Database](#)

## Functions

[~AcMapOSEConnectionReactor](#) [AcMapOSEConnectionReactor](#)

[BeforeConnect](#)

[BeforeDisconnect](#)

[BeforeSchemaChange](#)

[Connected](#)

[Disconnected](#)

[SchemaChanged](#)

# AcMapOSEExport class

[Functions](#) [Related Classes](#) [See Also](#)

Manages exporting data to an Oracle database from a project drawing.

Include AdMapOracleExport.h

## Overviews

[Exporting data](#)

## Code samples

[Exporting to an Oracle Database \(sample\)](#)

## Functions

[~AcMapOSEExport](#) [AcMapOSEExport](#)

[ExportObjects](#)

[ExportObjectsAll](#)

[Init](#)

## Enumerations

[EExportOptions](#)

# AcMapOSEExportReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom export reactors.

Include AdMapOracleReactor.h.

To monitor and react to export events, subclass a custom reactor from AcMapOSEExportReactor, and then add an instance of it to your application using AcMapOSEConnection::AddExportReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Database](#)

## Functions

[~AcMapOSEExportReactor](#) [AcMapOSEExportReactor](#)

[BeforeObjectCached](#)

[BeforeObjectsExported](#)

[ObjectCached](#)

[ObjectRejected](#)

[ObjectsExported](#)

# AcMapOSEImport class

[Functions](#) [Related Classes](#) [See Also](#)

Manages importing objects from an Oracle database to the project drawing.

Uses the AcMapOSEQuery class to specify the objects to import.

Include AdMapOracleImport.h.

## Overviews

[Querying and importing data](#)

## Code samples

[Importing from an Oracle Database \(sample\)](#)

## Functions

[~AcMapOSEImport](#) [AcMapOSEImport](#)  
[Import](#)

# AcMapOSEImportReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom import reactors.

Include AdMapOracleReactor.h.

To monitor and react to import events, subclass a custom reactor from AcMapOSEImportReactor, and then add an instance of it to your application using AcMapOSEConnection::AddImportReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Database](#)

## Functions

[~AcMapOSEImportReactor](#) [AcMapOSEImportReactor](#)

[BeforeRecordImport](#)

[RecordImported](#)

[RecordRejected](#)

# AcMapOSEObject class

[Functions](#) [Related Classes](#)

Represents the AutoCAD and Oracle counterparts of a queried object, where the AutoCAD counterpart resides in the project drawing, and the Oracle counterpart resides in the Oracle database. You can initialize the object with the ID of either counterpart: an AutoCAD ID or an Oracle ID.

AcMapOSEObject functions add or remove the object's AutoCAD counterpart from the EditSet, and they provide information about the object, such as

- Whether the AutoCAD counterpart has been erased or modified.
- Whether the AutoCAD counterpart is in the EditSet.
- If the object has an AutoCAD counterpart in another user's EditSet, who the user is.
- The version number of the AutoCAD counterpart, and whether the Oracle and AutoCAD version numbers agree (that is, whether the object is "up to date").
- The object's feature type.

Include AdMapOracleIdentification.h.

## Overviews

[Oracle Spatial and AutoCAD IDs](#) [Getting corresponding IDs](#)

## Code samples

[Getting corresponding IDs \(sample\)](#)

## Functions

[~AcMapOSEObject](#)

[AcMapOSEObject](#)

[AddToEditSet](#)

[FeatureName](#)

[GetAcadID](#)

[GetOracleID](#)

[IsErased](#)

[IsInEditSet](#)

[IsModified](#)

[IsUpToDate](#)

[Init](#)

[RemoveFromEditSet](#)

[Version](#)

[WhoHasIt](#)

# AcMapOSEProject class

[Functions](#) [Related Classes](#)

Represents the drawing database of the project drawing. Lets you filter queried or new drawing objects, and add or remove objects from the EditSet.

Include AdMapOracleProject.h.

## Overviews

[Filtering objects](#)

## Code samples

[Filtering objects \(sample\)](#)

## Functions

[~AcMapOSEProject](#) [AcMapOSEProject](#)

[AddToEditSet](#)

[FilterNewObjects](#)

[FilterQueriedObjects](#)

[Init](#)

[RemoveFromEditSet](#)

## Enumerations

[EProjectOptions](#)

# AcMapOracleQuery class

[Functions](#) [Related Classes](#) [See Also](#)

Specifies the objects to import for the AcMapOracleImport class.

Include AcMapOracleQuery.h.

## Overviews

[Querying and importing data](#)

## Code samples

[Importing from an Oracle Spatial database \(sample\)](#)

## Functions

[~AcMapOracleQuery](#) [AcMapOracleQuery](#)

[ConvertToSqlString](#)

[Init](#)

[InitWithCurrent](#)

[Load](#)

[Save](#)

## AcMapOracleGetConnection global function

Gets the connection object.

Include AcMapOracleConnection.h.

```
AcMapOracleConnection *  
AcMapOracleGetConnection();
```

Returns the connection object.

Before you can do anything with Oracle Spatial data, you must get the AcMapOracleConnection object.

```
AcMapOracleConnection *pConnection = AcMapOracleGetConnection();
```

See [Managing the connection \(overview\)](#).

# AcMapOracleConnection class

[Functions](#) [Related Classes](#) [See Also](#)

Manages the Oracle Spatial database connection and associated reactors.

Instantiated automatically.

Accessed by `AcMapOracleGetConnection()`, a global function.

Include `AcMapOracleConnection.h`.

## Overviews

[Managing the connection](#) [Managing reactors](#)

## Code samples

[Connecting to an Oracle Spatial database](#)

## Functions

[~AcMapOracleConnection](#)

[AddConnectionReactor](#)

[AddExportReactor](#)

[AddImportReactor](#)

[Connect](#)

[Database](#)

[Disconnect](#)

[IsConnected](#)

[IsSchemaValid](#)

[RemoveConnectionReactor](#)

[RemoveExportReactor](#)

[RemoveImportReactor](#)

[Schema](#)

[Service](#)

[UserName](#)

# AcMapOracleConnectionReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom connection reactors.

Include AcMapOracleReactor.h.

To monitor and react to connection events, subclass a custom reactor from AcMapOracleConnectionReactor, and then add an instance of it to your application using AcMapOracleConnection::AddConnectionReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Spatial database](#)

## Functions

[~AcMapOracleConnectionReactor](#) [AcMapOracleConnectionReactor](#)

[BeforeConnect](#)

[BeforeDisconnect](#)

[BeforeSchemaChange](#)

[Connected](#)

[Disconnected](#)

[SchemaChanged](#)

# AcMapOracleExport class

[Functions](#) [Related Classes](#) [See Also](#)

Manages exporting data to an Oracle Spatial database from a project drawing.

Include AcMapOracleExport.h

## Overviews

[Exporting data](#)

## Code samples

[Exporting to an Oracle Spatial database \(sample\)](#)

## Functions

[~AcMapOracleExport](#) [AcMapOracleExport](#)

[ExportObjects](#)

[ExportObjectsAll](#)

## Enumerations

[EExportOptions](#) Enum

# AcMapOracleExportReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom export reactors.

Include AcMapOracleReactor.h.

To monitor and react to export events, subclass a custom reactor from AcMapOracleExportReactor, and then add an instance of it to your application using AcMapOracleConnection::AddExportReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Spatial database](#)

## Functions

[~AcMapOracleExportReactor](#) [AcMapOracleExportReactor](#)

[BeforeObjectCached](#)

[BeforeObjectsExported](#)

[ObjectCached](#)

[ObjectRejected](#)

[ObjectsExported](#)

# AcMapOracleIdentification class

[Functions](#) [Related Classes](#)

For imported entities, uses the AutoCAD ID (AcDbObjectId) to get the Oracle ID, and vice versa.  
Include AcMapOracleIdentification.h.

## Overviews

[Oracle Spatial and AutoCAD IDs](#) [Getting corresponding IDs](#)

## Code samples

[Getting corresponding IDs \(sample\)](#)

## Functions

[~AcMapOracleIdentification](#)

[AcMapOracleIdentification](#)

[GetAcadID](#)

[GetOracleID](#)

[Init](#)

# AcMapOracleImport class

[Functions](#) [Related Classes](#) [See Also](#)

Manages importing data from an Oracle Spatial database to a project drawing.

Uses the AcMapOracleQuery class to specify the objects to import.

Include AcMapOracleImport.h.

## Overviews

[Querying and importing data](#)

## Code samples

[Importing from an Oracle Spatial database \(sample\)](#)

## Functions

[~AcMapOracleImport](#) [AcMapOracleImport](#)

[CheckImport](#)

[Import](#)

# AcMapOracleImportReactor class

[Functions](#) [Related Classes](#) [See Also](#)

Base class for custom import reactors.

Include AcMapOracleReactor.h.

To monitor and react to import events, subclass a custom reactor from AcMapOracleImportReactor, and then add an instance of it to your application using AcMapOracleConnection::AddImportReactor().

## Overviews

[Managing reactors](#)

## Code samples

[Connecting to an Oracle Spatial database](#)

## Functions

[~AcMapOracleImportReactor](#) [AcMapOracleImportReactor](#)

[BeforeRecordImport](#)

[RecordImported](#)

[RecordRejected](#)

Links

[AcMapEntityCreationSettings Class](#), [AcMapEntityCreationSettings Class](#)  
[AcMapEntityCreationSettings:: ~AcMapEntityCreationSettings Destructor](#)  
[AcMapEntityCreationSettings Class](#) | [AcMapEntityCreationSettings Class](#)

Destroys an instance of this class.

```
virtual ~AcMapEntityCreationSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapFloodParameters Class](#), [AcMapFloodParameters Class](#)

[AcMapFloodParameters:: ~AcMapFloodParameters Destructor](#)

[AcMapFloodParameters Class](#) | [AcMapFloodParameters Class](#)

Destroys an instance of this class.

```
virtual ~AcMapFloodParameters();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

[AcMapMarkerStyles:: ~AcMapMarkerStyles Destructor](#)

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Destroys an instance of this class.

```
virtual ~AcMapMarkerStyles();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

[AcMapMarkerStyles:: SetMarkerSize Method](#)

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Sets the error-marker size.

```
AcMap::EErrCode SetMarkerSize(  
    int newSize  
);
```

Parameters	Description
newSize	Input marker size.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

AcMapMarkerStyles:: SetTopoDuplicateCentroidMarker Method

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Sets the marker shape and color for a duplicate-centroid error.

```
AcMap::EErrorCode SetTopoDuplicateCentroidMarker(  
    ETopologyMarkType type,  
    int nColor  
);
```

Parameters	Description
type	Input ETopologyMarkTypeshape type.
nColor	Input color index.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

AcMapMarkerStyles:: SetTopoIncompleteAreaMarker Method

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Sets the marker shape and color for an incomplete-area error.

```
AcMap::EErrorCode SetTopoIncompleteAreaMarker(  
    ETopologyMarkType type,  
    int nColor  
);
```

Parameters	Description
type	Input ETopologyMarkTypeshape type.
nColor	Input color index.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

AcMapMarkerStyles:: SetTopoIntersectionMarker Method

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Sets the marker shape and color for an intersection error.

```
AcMap::EErrorCode SetTopoIntersectionMarker(  
    ETopologyMarkType type,  
    int nColor  
);
```

Parameters	Description
type	Input ETopologyMarkTypeshape type.
nColor	Input color index.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMarkerStyles Class](#), [AcMapMarkerStyles Class](#)

AcMapMarkerStyles:: SetTopoMissingCentroidMarker Method

[AcMapMarkerStyles Class](#) | [AcMapMarkerStyles Class](#)

Sets the marker shape and color for a missing-centroid error.

```
AcMap::EErrorCode SetTopoMissingCentroidMarker(  
    ETopologyMarkType type,  
    int nColor  
);
```

Parameters	Description
type	Input ETopologyMarkTypeshape type.
nColor	Input color index.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapNetAnalysisParameters Class](#), [AcMapNetAnalysisParameters Class](#)  
AcMapNetAnalysisParameters:: ~AcMapNetAnalysisParameters Destructor  
[AcMapNetAnalysisParameters Class](#) | [AcMapNetAnalysisParameters Class](#)

Destroys an instance of this class.

```
virtual ~AcMapNetAnalysisParameters();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapPointCreationSettings Class](#), [AcMapPointCreationSettings Class](#)  
AcMapPointCreationSettings::~~AcMapPointCreationSettings Destructor  
[AcMapPointCreationSettings Class](#) | [AcMapPointCreationSettings Class](#)

Destroys an instance of this class.

```
virtual ~AcMapPointCreationSettings();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElement Class](#), [AcMapTopoElement Class](#)

[AcMapTopoElement:: ~AcMapTopoElement Destructor](#)

[AcMapTopoElement Class](#) | [AcMapTopoElement Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElement Class](#), [AcMapTopoElement Class](#)

AcMapTopoElement:: AcMapTopoElement Constructor

[AcMapTopoElement Class](#) | [AcMapTopoElement Class](#)

Constructs an instance of this class.

```
AcMapTopoElement();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElement Class](#), [AcMapTopoElement Class](#)

AcMapTopoElement:: GetEntity Method

[AcMapTopoElement Class](#) | [AcMapTopoElement Class](#)

Retrieves the object identifier of the AutoCAD entity associated with this object.

```
virtual AcMap::EErrCode GetEntity(  
    AcDbObjectId& featureID  
) const = 0;
```

Parameters	Description
featureID	Output the feature's ID.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElement Class](#), [AcMapTopoElement Class](#)

[AcMapTopoElement:: GetID Method](#)

[AcMapTopoElement Class](#) | [AcMapTopoElement Class](#)

Retrieves the unique identifier of this object in its topology.

```
virtual long GetID() const = 0;
```

Returns

Returns the ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoElement Class](#), [AcMapTopoElement Class](#)

[AcMapTopoElement::IsOnThisObject Method](#)

[AcMapTopoElement Class](#) | [AcMapTopoElement Class](#)

Determines whether a specified point is within tolerance of this object.

```
virtual bool IsOnThisObject(  
    const AcGePoint3d& point,  
    double dTolerance  
) const = 0;
```

Parameters	Description
point	Input point to test.
dTolerance	Input the maximum distance to the point (ignored for polygons). The measurement units depend on the current drawing's unit settings.

## Returns

Returns true if the point is within tolerance of this object, or false if it is not.

## Remarks

If this object is a polygon, then the point must be inside or on the boundary, regardless of tolerance.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElementPtrArray Template](#), [AcMapTopoElementPtrArray Template](#)  
AcMapTopoElementPtrArray:: ~AcMapTopoElementPtrArray Destructor  
[AcMapTopoElementPtrArray Template](#) | [AcMapTopoElementPtrArray Template](#)

Destroys the array.

`~AcMapTopoElementPtrArray();`

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoElementPtrArray Template](#), [AcMapTopoElementPtrArray Template](#)

AcMapTopoElementPtrArray:: Empty Method

[AcMapTopoElementPtrArray Template](#) | [AcMapTopoElementPtrArray Template](#)

Releases memory allocated to the array.

```
void Empty();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge:: EDirection Enumeration

[AcMapTopoFullEdge Class](#)

Enumerates the directions of a full edge.

```
enum EDirection {  
    eBiDir = 0,  
    eForward = 1,  
    eBackward = -1  
};
```

File

AcMapTopoFullEdge.h

Parameters	Description
eBiDir	Bidirectional.
eForward	Direction from start node to end node.
eBackward	Direction from end node to start node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)  
AcMapTopoFullEdge:: ~AcMapTopoFullEdge Destructor  
[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoFullEdge();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: GetDirection Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the direction of this edge.

```
EDirection GetDirection();
```

Returns

Returns the EDirectionedge direction.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge:: GetEntity Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the object identifier of the AutoCAD entity associated with this edge.

```
virtual AcMap::EErrCode GetEntity(  
    AcDbObjectId& featureID  
) const;
```

Parameters	Description
featureID	Output the feature's ID.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge::GetHalfEdge Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the half edge that traverses one way along this full edge.

```
AcMap::EErrorCode GetHalfEdge(  
    AcMapTopoHalfEdge*& pHalfEdge,  
    bool bOnLeft  
);
```

Parameters	Description
pHalfEdge	Output AcMapTopoHalfEdgehalf edge. This value points to a copy of the half edge referenced by the half edge object. The caller is responsible for freeing this object.
bOnLeft	Input true to choose the left edge, or false to choose the right edge.

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::EErrorCode kErrTopTraceLinkNotExist if no edge is found.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: GetID Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the identifier of this edge in its topology.

```
virtual long GetID() const;
```

Returns

Returns the ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: GetLength Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the length of this edge.

**double** GetLength();

Returns

Returns the edge length.

Remarks

The measurement units depend on the current drawing's unit settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: GetNextEdge Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the next edge by moving forward or backward along this edge and turning left or right at the next node.

```
AcMap::EErrCode GetNextEdge(  
    AcMapTopoFullEdge*& pOutGoingEdge,  
    bool bGoForward,  
    bool bGoLeft  
);
```

Parameters	Description
pOutGoingEdge	Output next edge. This value points to a copy of the full edge referenced by the full edge object. The caller is responsible for freeing this object.
bGoForward	Input true to go forward, or false to go backward.
bGoLeft	Input true to turn left, or false to turn right.
Returns	

Returns AcMap::EErrCode kOk if successful. Returns AcMap::EErrCode kErrTopTraceLinkNotExist if no edge is found.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge:: GetNextNode Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the next node by moving forward or backward along this edge.

```
AcMap::EErrorCode GetNextNode(  
    AcMapTopoNode*& pNextNode,  
    bool bGoForward  
);
```

Parameters	Description
pNextNode	Output next AcMapTopoNodenode. This value points to a copy of the node referenced by the node object. The caller is responsible for freeing this object.
bGoForward	Input true to go forward, or false to go backward.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge::GetPolygon Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the polygon associated with this edge (default level).

```
AcMap::EErrorCode GetPolygon(  
    AcMapTopoPolygon*& pPolygon,  
    bool bOnLeft  
);
```

Parameters	Description
pPolygon	Output AcMapTopoPolygonpolygon. This value points to a copy of the polygon referenced by the polygon object. The caller is responsible for freeing this object.
bOnLeft	Input true to choose the left polygon, or false to choose the right polygon.

Returns

Returns AcMap::EErrorCode kOk if successful.. Returns AcMap::EErrorCode kErrTopNotExist if no polygon is associated with this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge:: GetResistance Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the resistance of this edge.

```
double GetResistance(  
    bool bGoForward  
);
```

Parameters

Description

bGoForward

Input true to go forward, or false to go backward.

Returns

Returns the edge resistance.

Remarks

The measurement units depend on your application. The default resistance is the length of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge::GetRing Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the ring associated with this edge (default level).

```
AcMap::EErrorCode GetRing(  
    AcMapTopoRing*& pRing,  
    bool bOnLeft  
);
```

Parameters	Description
pRing	Output AcMapTopoRingring. This value points to a copy of the ring referenced by the ring object. The caller is responsible for freeing this object.
bOnLeft	Input true to choose the left edge, or false to choose the right edge.

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::EErrorCode kErrTopNotExist if no ring is associated with this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: GetTopology Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Retrieves the topology that contains this edge.

```
AcMap::EErrorCode GetTopology(  
    const AcMapTopology*& pTopology  
);
```

Parameters	Description
pTopology	Output AcMapTopologytopology reference. Do not delete this reference.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: IsOnThisObject Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Determines whether a specified point is within tolerance of this edge.

```
virtual bool IsOnThisObject(  
    const AcGePoint3d& point,  
    double dTolerance  
) const;
```

Parameters	Description
point	Input point to test.
dTolerance	Input maximum distance to the point.

Returns

Returns true if the point is within tolerance of this edge, or false if it is not.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

[AcMapTopoFullEdge:: SetDirection Method](#)

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Sets the direction of this edge.

```
AcMap::EErrorCode SetDirection(  
    EDirection eDir  
);
```

Parameters	Description
eDir	Input EDirectiondirection to set.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoFullEdge Class](#), [AcMapTopoFullEdge Class](#)

AcMapTopoFullEdge:: SetResistance Method

[AcMapTopoFullEdge Class](#) | [AcMapTopoFullEdge Class](#)

Sets the resistance of this edge.

```
AcMap::EErrorCode SetResistance(  
    bool bGoForward,  
    double dResistance  
);
```

Parameters	Description
bGoForward	Input true to go forward, or false to go backward.
dResistance	Input resistance to set.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Remarks

The measurement units depend on your application. The default resistance is the length of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
AcMapTopoHalfEdge:: ~AcMapTopoHalfEdge Destructor  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoHalfEdge();  
Returns
```

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
[AcMapTopoHalfEdge:: GetFullEdge Method](#)  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the full edge that contains this half edge.

```
AcMap::EErrorCode GetFullEdge(  
    const AcMapTopoFullEdge*& pFullEdge  
);
```

Parameters	Description
pFullEdge	Output AcMapTopoFullEdgefull edge reference. Do not delete this reference.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)

[AcMapTopoHalfEdge:: GetNextEdge Method](#)

[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the next edge by turning left or right at the next node.

```
AcMap::EErrorCode GetNextEdge(  
    AcMapTopoHalfEdge*& pOutGoingEdge,  
    bool bGoLeft  
);
```

Parameters	Description
pOutGoingEdge	Output next edge. This value points to a copy of the half edge referenced by the half edge object. The caller is responsible for freeing this object.
bGoLeft	Input true to turn left, or false to turn right.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)

[AcMapTopoHalfEdge:: GetNextNode Method](#)

[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the next node by moving forward along this edge.

```
AcMap::EErrorCode GetNextNode(  
    AcMapTopoNode*& pNextNode  
);
```

Parameters	Description
pNextNode	Output next AcMapTopoNodenode. This value points to a copy of the node referenced by the node object. The caller is responsible for freeing this object.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)

[AcMapTopoHalfEdge:: GetPolygon Method](#)

[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the polygon associated with this edge (default level).

```
AcMap::EErrorCode GetPolygon(  
    AcMapTopoPolygon*& pPolygon  
);
```

### Parameters

### Description

pPolygon

Output AcMapTopoPolygonpolygon. This value points to a copy of the polygon referenced by the polygon object. The caller is responsible for freeing this object.

### Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
[AcMapTopoHalfEdge:: GetPreviousNode Method](#)  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the previous node by moving backward along this edge.

```
AcMap::EErrorCode GetPreviousNode(  
    AcMapTopoNode*& pPreviousNode  
);
```

Parameters	Description
pPreviousNode	Output previous AcMapTopoNodenode. This value points to a copy of the node referenced by the node object. The caller is responsible for freeing this object.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)

[AcMapTopoHalfEdge:: GetResistance Method](#)

[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the resistance of this edge.

```
double GetResistance();
```

Returns

Returns the edge resistance.

Remarks

The measurement units depend on your application. The default resistance is the length of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
[AcMapTopoHalfEdge:: GetRing Method](#)  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the ring associated with this edge (default level).

```
AcMap::EErrorCode GetRing(  
    AcMapTopoRing*& pRing  
);
```

Parameters	Description
pRing	Output AcMapTopoRingring. This value points to a copy of the ring referenced by the ring object. The caller is responsible for freeing this object.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
[AcMapTopoHalfEdge:: GetTopology Method](#)  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Retrieves the topology that contains this edge.

```
AcMap::EErrorCode GetTopology(  
    const AcMapTopology*& pTopology  
);
```

Parameters	Description
pTopology	Output AcMapTopologytopology reference. Do not delete this reference.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoHalfEdge Class](#), [AcMapTopoHalfEdge Class](#)  
AcMapTopoHalfEdge:: SetResistance Method  
[AcMapTopoHalfEdge Class](#) | [AcMapTopoHalfEdge Class](#)

Sets the resistance of this edge.

```
AcMap::EErrorCode SetResistance(  
    double dResistance  
);
```

Parameters	Description
dResistance	Input resistance to set.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Remarks

The measurement units depend on your application. The default resistance is the length of this edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

[AcMapTopoIterator:: ~AcMapTopoIterator Destructor](#)

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

[AcMapTopoIterator:: AcMapTopoIterator Constructor](#)

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Constructs an instance of this class.

```
AcMapTopoIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

[AcMapTopoIterator:: Count Method](#)

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Counts the number of elements in the iteration.

```
int Count();
```

Returns

Returns the number of elements.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator:: First Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Moves to the first element in the iteration.

```
AcMap::EErrorCode First();
```

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator:: GetDescription Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Retrieves the description of this topology model.

```
const ACHAR* GetDescription() const;
```

Returns

Returns the description of the topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator:: GetName Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Retrieves the name of this topology model.

```
const ACHAR* GetName() const;
```

Returns

Returns the name of the topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

[AcMapTopoIterator:: GetTopology Method](#)

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Retrieves the topology of the current element in the iteration.

```
AcMap::EErrorCode GetTopology(  
    AcMapTopology*& pTopology  
);
```

### Parameters

### Description

pTopology

Output AcMapTopologytopology. This value points to a copy of the topology referenced by the topology object. The caller is responsible for freeing this object.

### Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator::GetType Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Retrieves the type of this topology model.

```
ETopologyType GetType() const;
```

Returns

Returns the ETopologyType of the topology model. The topology is ePoint|eFixed for a point topology, eLinear|eFixed for a network topology, or ePolygon|eFixed for a polygon topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator:: IsDone Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Determines whether the iterator has reached the end of the collection.

**bool** IsDone();

Returns

Returns true if the iterator has reached the end of the collection; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoIterator Class](#), [AcMapTopoIterator Class](#)

AcMapTopoIterator:: Next Method

[AcMapTopoIterator Class](#) | [AcMapTopoIterator Class](#)

Advances to the next element in the iteration.

```
AcMap::EErrorCode Next();
```

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#)

AcMapTopology:: EAuditResults Enumeration

[AcMapTopology Class](#)

Enumerates the result types of an audit.

```
enum EAuditResults {  
    eIncorrect = 0x0001,  
    eIncomplete = 0x0002  
};
```

File

AcMapTopology.h

Parameters	Description
eIncorrect	The topology is incorrect.
eIncomplete	The topology is incomplete.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#)

AcMapTopology:: ECreateOptions Enumeration

[AcMapTopology Class](#)

Enumerates the options for creating a topology.

```
enum ECreateOptions {  
    eStopAtMultipleCentroid = 0x0001,  
    eIgnoreIncompleteArea = 0x0002,  
    eHighlightErrors = 0x0004,  
    eUsePersistentMarkers = 0x0008,  
    eHighlightSliverPolygons = 0x0010  
};
```

File

AcMapTopology.h

Parameters	Description
eStopAtMultipleCentroid	Stop at multiple centroid.
eIgnoreIncompleteArea	Ignore an incomplete area.
eHighlightErrors	Highlight creation errors.
eUsePersistentMarkers	Use persistent markers.
eHighlightSliverPolygons	Highlight silver polygons.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#)

AcMapTopology:: EOpenMode Enumeration

[AcMapTopology Class](#)

Enumerates the ways to open a topology model.

```
enum EOpenMode {  
    eForRead = 0,  
    eForWrite = 1  
};  
File
```

AcMapTopology.h

Parameters	Description
eForRead	Open for read.
eForWrite	Open for write.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#)

AcMapTopology:: EStatus Enumeration

[AcMapTopology Class](#)

Enumerates the status types of a topology model.

```
enum EStatus {  
    eClosed,  
    eOpenForRead,  
    eOpenForWrite  
};  
File
```

AcMapTopology.h

Parameters	Description
eClosed	Closed.
eOpenForRead	Open for read.
eOpenForWrite	Open for write.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)  
AcMapTopology:: ~AcMapTopology Destructor  
[AcMapTopology Class](#) | [AcMapTopology Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopology();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::AddCurveObject Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Adds a curve (linear object) to this topology model.

```
AcMap::EErrorCode AddCurveObject(  
    AcMapTopoFullEdge*& pFullEdge,  
    const AcDbObjectId& curveId  
);
```

Parameters	Description
pFullEdge	Output created AcMapTopoFullEdgeedge. This value points to a copy of the full edge referenced by the input curve. The caller is responsible for freeing this object.
curveId	Input curve to create the edge on.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::AddPointObject Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Adds a point to this topology model.

```
AcMap::EErrorCode AddPointObject(  
    AcMapTopoNode*& pNode,  
    const AcDbObjectId& pointId  
);
```

Parameters	Description
pNode	Output created AcMapTopoNodenode. This value points to a copy of the node referenced by the specified point. The caller is responsible for freeing this object.
pointId	Input ID of the point to create the node on.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::AddPolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Adds objects to a polygon topology model.

```
AcMap::EErrorCode AddPolygons(  
    const AcDbObjectIdArray& aObjIds  
);
```

Parameters	Description
aObjIds	Input array of IDs of edge objects.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

The added objects must be linear AutoCAD entities, and must not intersect with each other or with the polygon topology being added to.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: SetNodeCreationSettings Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Sets the node-creation parameters of this topology model.

```
AcMap::EErrorCode SetNodeCreationSettings(  
    const AcMapPointCreationSettings& nodeCreationSettings  
);
```

Parameters	Description
nodeCreationSettings	Input <a href="#">AcMapPointCreationSettings</a> node-creation settings.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::SetEdgeCreationSettings Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Sets the edge-creation parameters of this topology model.

```
AcMap::EErrorCode SetEdgeCreationSettings(  
    const AcMapEntityCreationSettings& edgeCreationSettings  
);
```

Parameters	Description
edgeCreationSettings	Input <a href="#">AcMapEntityCreationSettings</a> edge-creation settings.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::SetCentroidCreationSettings Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Sets the centroid-creation parameters of this topology model.

```
AcMap::EErrorCode SetCentroidCreationSettings(  
    const AcMapPointCreationSettings& centroidCreationSettings  
);
```

Parameters	Description
centroidCreationSettings	Input <a href="#">AcMapPointCreationSettings</a> centroid-creation settings.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: Close Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Closes this topology model.

```
AcMap::EErrorCode Close();
```

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: Create Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Creates a topology. To set topology-creation values, call the three functions [SetNodeCreationSettings\(\)](#), [SetEdgeCreationSettings\(\)](#), and [SetCentroidCreationSettings\(\)](#). If caller does not call these functions, then node creation defaults to false, missing-centroid creation defaults to true, and the node or centroid block name defaults to ACAD\_POINT.

```
AcMap::EErrCode Create(  
    const AcDbObjectIdArray& edges,  
    const AcDbObjectIdArray& nodes,  
    const AcDbObjectIdArray& centroids,  
    int nType,  
    int nCreateOptions = eStopAtMultipleCentroid | eIgnoreIncomplete  
    double dTolerance = 0.01  
);
```

Parameters	Description
edges	Input edge object IDs.
nodes	Input node object IDs.
centroids	Input centroid object IDs.
nType	Input ETopologyType topology type.
nCreateOptions	Input ECreateOption topology creation options. The default value is eStopAtMultipleCentroid   eIgnoreIncompleteArea   eUsePersistentMarkers.
dTolerance	Input topology-creation tolerance. The default value is 0.01.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: Create Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Creates a topology - this function is not yet implemented in AutoCAD Map; use the other Create()function.

```
AcMap::EErrCode Create(  
    const AcDbObjectIdArray& geometry,  
    int nType,  
    int nCreateOptions = eStopAtMultipleCentroid | eIgnoreIncomplete  
    double dTolerance = 0.01  
);
```

Parameters	Description
geometry	Input geometry array.
nType	Input ETopologyType topology type.
nCreateOptions	Input ECreateOption topology creation options. The default value is eStopAtMultipleCentroid   eIgnoreIncompleteArea   eUsePersistentMarkers.
dTolerance	Input topology-creation tolerance. The default value is 0.01.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::DeleteEdge Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Deletes an edge from this topology model.

```
AcMap::EErrorCode DeleteEdge(  
    long lEdgeID  
);
```

Parameters	Description
lEdgeID	Input ID of the edge to delete.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

If the edge is in a polygon, that polygon becomes broken. Before deleting an edge, open the topology should for writing and free all references associated with the edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::DeleteNode Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Deletes a node from this topology model.

```
AcMap::EErrCode DeleteNode(  
    const AcMapTopoNode& node  
);
```

### Parameters

### Description

node

Input object reference of the AcMapTopoNodenode to delete.

### Returns

Returns AcMap::EErrCode kOk if successful. Returns AcMap::EErrCode kErrNotClosed if the node object is open for read/write. Returns AcMap::EErrCode kErrTopDeleteDisabled if the node is not a pseudonode for a linear or polygon topology.

### Remarks

Only current stand-alone nodes can be deleted. For a linear or polygon topology, only a pseudonode can be deleted. Before deleting a node, open a topology for writing. After the deletion succeeds, the object reference passed in the node parameter becomes invalid; the caller is responsible for freeing this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::DeleteNode Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Deletes a node from this topology model.

```
AcMap::EErrCode DeleteNode(  
    long lNodeID  
);
```

Parameters	Description
lNodeID	Input ID of the node to delete.

Returns

Returns AcMap::EErrCode kOk if successful. Returns AcMap::EErrCode kErrNotClosed if the node object is open for read/write. Returns AcMap::EErrCode kErrTopDeleteDisabled if the node is not a pseudonode for a linear or polygon topology.

## Remarks

Only current stand-alone nodes can be deleted. For a linear or polygon topology, only a pseudonode can be deleted. Before deleting a node, open the topology for writing and free all references associated with the node. After the deletion succeeds, the node object referenced by lNodeID becomes invalid; the caller is responsible for freeing this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::DeletePolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Deletes a polygon from the current topology model.

```
AcMap::EErrorCode DeletePolygon(  
    const AcMapTopoPolygon& polygon  
);
```

Parameters	Description
polygon	Input object reference of the AcMapTopoPolygonpolygon to delete.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

Before deleting the polygon, open the topology for writing. After the deletion succeeds, the object reference passed in the polygon parameter becomes invalid; the caller is responsible for freeing this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::DeletePolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Deletes a polygon from the current topology model.

```
AcMap::EErrorCode DeletePolygon(  
    long lPolygongID  
);
```

Parameters	Description
lPolygongID	Input ID of the polygon to delete.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

Before deleting a polygon, open the topology for writing and free all references associated with the polygon. After the deletion succeeds, the object reference with ID lPolygongID becomes invalid; the caller is responsible for freeing this object.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::FindEdge Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the edge closest to a specified point.

```
AcMap::EErrCode FindEdge(  
    AcMapTopoFullEdge*& pEdge,  
    double* pDistance,  
    const AcGePoint3d& point  
);
```

Parameters	Description
pEdge	Output AcMapTopoFullEdgeedge object. This value points to a copy of the full edge referenced by the found full edge. The caller is responsible for freeing this object.
pDistance	Output distance from the point to the nearest edge. The measurement units depend on the current drawing's unit settings.
point	Input point.
Returns	

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

## Remarks

In AutoCAD Map, the specified point must be on the edge; the returned distance always is 0.0.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::FindEdge Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the edge with the specified ID.

```
AcMap::EErrorCode FindEdge(  
    AcMapTopoFullEdge*& pEdge,  
    long lId  
);
```

Parameters

Description

pEdge

Output AcMapTopoFullEdgeedge object. This value points to a copy of the full edge referenced by the found full edge. The caller is responsible for freeing this object.

lId

Input ID of the edge to find.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::FindNeighborPolygons Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds all polygons that are neighbors of those edges that are based on a specified curve.

```
AcMap::EErrorCode FindNeighborPolygons(  
    AcMapPolygonPtrArray& apPolygons,  
    AcDbObjectId curveId  
);
```

Parameters	Description
apPolygons	Output array of the neighbor polygons.
curveId	Input ID of a curve in the topology.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::FindNode Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the node nearest a specified point.

```
AcMap::EErrorCode FindNode(  
    AcMapTopoNode*& pNode,  
    double* pDistance,  
    const AcGePoint3d& point  
);
```

Parameters	Description
pNode	Output AcMapTopoNodenode object. This value points to a copy of the node referenced by the found node. The caller is responsible for freeing this object.
pDistance	Output distance from the point to the nearest node. The measurement units depend on the current drawing's unit settings.
point	Input point.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::FindPolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the smallest polygon containing a specified point.

```
AcMap::EErrorCode FindPolygon(  
    AcMapTopoPolygon*& pPolygon,  
    const AcGePoint3d& point  
);
```

Parameters	Description
pPolygon	Output AcMapTopoPolygonpolygon object. This value points to a copy of the polygon referenced by the found polygon. The caller is responsible for freeing this object.
point	Input point.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)  
AcMapTopology::FindTopologyObject Method  
[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the object (node, full edge, or polygon) nearest a specified point within a specified tolerance.

```
AcMap::EErrCode FindTopologyObject(  
    AcMapTopoElement*& pElement,  
    const AcGePoint3d& point,  
    double dTolerance  
);
```

Parameters	Description
pElement	Output closest AcMapTopoElementobject. This value points to a copy of the node, full edge, or polygon referenced by the found object. The caller is responsible for freeing this object.
point	Input point.
dTolerance	Input tolerance.
Returns	

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

## Remarks

If a node is within tolerance, then it is returned. Otherwise, if an edge is within tolerance, then it is returned. Otherwise, the smallest enclosing polygon is returned. If no objects are within tolerance, NULL is returned.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::GetBackwardEdge Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the half edge with the specified object ID from this topology model.

```
AcMap::EErrorCode GetBackwardEdge(  
    AcMapTopoHalfEdge*& pHalfEdge,  
    long lFullEdgeId  
);
```

Parameters	Description
pHalfEdge	Output AcMapTopoHalfEdgehalf edge object. This value points to a copy of the half edge referenced by the specified half edge object ID. The caller is responsible for freeing this object.
lFullEdgeId	Input full edge ID.

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::kErrWrongType if the topology is a polygon. Returns AcMap::kErrBadInput if the full edge ID is invalid.

Remarks

The direction of the half edge is from the end node to the start node of the full edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: GetDescription Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the description of this topology model.

```
const ACHAR* GetDescription() const;
```

Returns

Returns the description of the topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: GetEntityId Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the object ID of the AutoCAD entity associated with the specified topology element ID in the current drawing or source drawing.

```
AcMap::EErrorCode GetEntityId(  
    AcDbObjectId& aObjId,  
    long lElementId  
);
```

Parameters	Description
aObjId	Output object ID of the AutoCAD entity.
lElementId	Input topology element ID.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: GetEntityIds Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the object IDs of all the AutoCAD entities associated with the topology elements in the current drawing.

```
AcMap::EErrorCode GetEntityIds(  
    AcDbObjectIdArray& aObjIds  
);
```

Parameters	Description
aObjIds	Output array of AutoCAD entity object IDs.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::GetForwardEdge Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the half edge with the specified object ID from this topology model.

```
AcMap::EErrorCode GetForwardEdge(  
    AcMapTopoHalfEdge*& pHalfEdge,  
    long lFullEdgeId  
);
```

Parameters	Description
pHalfEdge	Output AcMapTopoHalfEdgehalf edge object. This value points to a copy of the half edge referenced by the specified half edge object ID. The caller is responsible for freeing this object.
lFullEdgeId	Input full edge ID.

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::kErrWrongType if the topology is not a polygon. Returns AcMap::kErrBadInput if the full edge ID is invalid.

Remarks

The direction of the half edge is from the start node to the end node of the full edge.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::GetFullEdge Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the full edge with the specified object ID from this topology model.

```
AcMap::EErrorCode GetFullEdge(  
    AcMapTopoFullEdge*& pFullEdge,  
    long lId  
);
```

Parameters	Description
pFullEdge	Output AcMapTopoFullEdgefull edge object. This value points to a copy of the full edge referenced by the specified full edge object ID. The caller is responsible for freeing this object.
lId	Input full edge object ID.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::GetFullEdges Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves all full edges from this topology model.

```
AcMap::EErrorCode GetFullEdges(  
    AcMapFullEdgePtrArray& apFullEdges  
);
```

Parameters	Description
apFullEdges	Output array of full edges.

Returns

Returns `AcMap::EErrorCode kOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: GetMarkerStyles Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the pointer to the error-marker settings of this topology model.

```
AcMapMarkerStyles* GetMarkerStyles();
```

Returns

Returns a pointer to the AcMapMarkerStyleserror-marker settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: GetName Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the name of this topology model.

```
const ACHAR* GetName() const;
```

Returns

Returns the name of the topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::GetNode Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the node with the specified object ID from this topology model.

```
AcMap::EErrorCode GetNode(  
    AcMapTopoNode*& pNode,  
    long lId  
);
```

Parameters	Description
pNode	Output AcMapTopoNodenode object. This value points to a copy of the node referenced by the specified node object ID. The caller is responsible for freeing this object.
lId	Input node object ID.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: GetNodes Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves all nodes from this topology model.

```
AcMap::EErrorCode GetNodes(  
    AcMapNodePtrArray& apNodes  
);
```

Parameters	Description
apNodes	Output array of nodes.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::GetPolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the polygon with the specified object ID from this topology model.

```
AcMap::EErrorCode GetPolygon(  
    AcMapTopoPolygon*& pPolygon,  
    long lId  
);
```

Parameters	Description
pPolygon	Output AcMapTopoPolygonpolygon object. This value points to a copy of the polygon referenced by the specified polygon object ID. The caller is responsible for freeing this object.
lId	Input polygon object ID.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::GetPolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves all polygons from this topology model.

```
AcMap::EErrorCode GetPolygons(  
    AcMapPolygonPtrArray& apPolygons  
);
```

Parameters	Description
apPolygons	Output array of polygons.

Returns

Returns `AcMap::EErrorCode kOk` if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: GetStatus Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the status of this topology model.

```
EStatus GetStatus() const;
```

Returns

Returns the EStatusstatus of the topology model.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::GetType Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Retrieves the type of this topology model.

```
ETopologyType GetType() const;
```

Returns

Returns the ETopologyType of the topology model.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: IsComplete Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model is complete.

```
bool IsComplete() const;
```

Returns

Returns true if the topology model is complete; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::IsFixedType Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model can contain only elements of its own type.

```
bool IsFixedType() const;
```

Returns

Returns true if the topology model is fixed; otherwise, returns false.

Remarks

As currently implemented, this function always returns true. A non-fixed topology model (that is, one for which this function normally would return false) allows dangling edges in a polygon-only topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: IsLinearType Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model is a network topology.

**bool** IsLinearType() **const**;

Returns

Returns true if the topology model is a network topology; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::IsLogicalType Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether the elements in this topology model can be connected if their graphical representations are farther apart than the tolerance value.

```
bool IsLogicalType() const;
```

Returns

Returns true if the topology model is a logical topology; otherwise, returns false.

Remarks

As currently implemented, this function always returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: IsPointType Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model is a point topology.

```
bool IsPointType() const;
```

Returns

Returns true if the topology model is a point topology; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: IsPolygonType Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model is a polygon topology.

```
bool IsPolygonType() const;
```

Returns

Returns true if the topology model is a polygon topology; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::MergePolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Merges neighboring polygons, dissolving the boundaries between them.

```
AcMap::EErrCode MergePolygons(  
    AcArray<long>& aPolygonIds  
);
```

Parameters	Description
aPolygonIds	Input array of IDs of the polygons to merge.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

## Remarks

If one of the specified polygons shares no boundary with the others, this function fails. After a merge succeeds, the original polygon reference apPolygons[0] becomes invalid; the caller is responsible for retrieving the new polygon reference with ID apPolygons[0]. The other (now merged) polygons also become invalid and the caller is responsible for freeing them.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::MergePolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Merges neighboring polygons, dissolving the boundaries between them.

```
AcMap::EErrorCode MergePolygons(  
    AcMapPolygonPtrArray& apPolygons  
);
```

Parameters	Description
apPolygons	Input array of polygons to merge.

Returns

Returns `AcMap::EErrorCode kOk` if successful; otherwise, returns a different error code.

## Remarks

If one of the specified polygons shares no boundary with the others, this function fails. After a merge succeeds, the `aPolygons[0]` will reference the newly merged polygon. The other polygon references become invalid and are removed from `aPolygons[]`.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::MergePolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Merges two neighboring polygons, dissolving the boundaries between them.

```
AcMap::EErrorCode MergePolygons(  
    const AcMapTopoPolygon& polygon1,  
    const AcMapTopoPolygon& polygon2  
);
```

Parameters	Description
polygon1	Input object reference of the first AcMapTopoPolygonpolygon to merge.
polygon2	Input object reference of the second AcMapTopoPolygonpolygon to merge.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

After the merge succeeds, polygon1 will point to the new, merged polygon object and polygon2 becomes invalid; the caller is responsible for freeing the object referenced by polygon2.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::MergePolygons Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Merges two neighboring polygons, dissolving the boundaries between them.

```
AcMap::EErrCode MergePolygons(  
    long lPolygonId1,  
    long lPolygonId2  
);
```

Parameters	Description
lPolygonId1	Input ID of the first polygon to merge.
lPolygonId2	Input ID of the second polygon to merge.

## Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

## Remarks

After a merge succeeds, the object reference lPolygonId1 becomes invalid; the caller is responsible for retrieving the new polygon reference from the polygon object with ID lPolygonId1. The object reference lPolygonId2 also becomes invalid and the caller is responsible for freeing the object referenced by lPolygonId2.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::MoveNode Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Moves a node to the new position.

```
AcMap::EErrCode MoveNode(  
    const AcMapTopoNode& node,  
    const AcGePoint3d& point  
);
```

Parameters	Description
node	Input object reference of the AcMapTopoNodenode to move.
point	Input point to move the node to.
Returns	

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::MoveNode Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Moves a node to the new position.

```
AcMap::EErrorCode MoveNode(  
    long lNodeID,  
    const AcGePoint3d& point  
);
```

Parameters	Description
lNodeID	Input ID of the node to move.
point	Input point to move the node to.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Remarks

After the move succeeds, the node object referenced by lNodeID becomes invalid; the caller must get the new object reference.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::NeedsRefresh Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Determines whether this topology model is correct.

**bool** NeedsRefresh() **const**;

Returns

Returns true if the topology model is incorrect; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: Open Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Opens the topology model in only the current drawing. After you are finished with a topology model, call Close()to close it.

```
AcMap::EErrorCode Open(  
    EOpenMode eMode  
);
```

Parameters	Description
eMode	Input EOpenModeopen mode.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology:: Open Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Opens the topology model in both current and source (attached) drawings. After you are finished with a topology model, call Close()to close it.

```
AcMap::EErrCode Open(  
    EOpenMode eMode,  
    bool bCreateObjects,  
    bool bSelectObjectsForSaveBack,  
    int* nAuditResults = NULL  
);
```

### Parameters

### Description

eMode

Input EOpenModeopen mode.

bCreateObjects

Input true to create objects from the source drawing; otherwise, false.

bSelectObjectsForSaveBack

Input true to select topology objects in the save set; otherwise, false.

nAuditResults

Input pointer that determines topology audit behavior. If NULL (the default value), no audit is performed. If the pointer is valid, the returned value zero indicates that the topology is correct and complete; otherwise, the returned value is a one of (or a combination of) the EAuditResultsvalues.

### Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: Refresh Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Updates a topology element.

```
AcMap::EErrorCode Refresh();
```

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::EErrorCode kErrTopOpenSourceDwgTopo if the topology is loaded from the source drawing. Otherwise, returns a different error code.

Remarks

The topology must be loaded from the current drawing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: ShowGeometry Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Highlights the topology model with the specified color.

```
AcMap::EErrorCode ShowGeometry(  
    int nColor  
);
```

Parameters	Description
nColor	Input color index.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology:: SplitPolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Splits a polygon along a curve.

```
AcMap::EErrCode SplitPolygon(  
    AcMapObjectPtrArray& apResults,  
    const AcDbObjectId& curveId,  
    const AcMapTopoPolygon& polygon  
);
```

Parameters	Description
apResults	Output new and affected topological objects.
curveId	Input curve to split at.
polygon	Input object reference of AcMapTopoPolygonpolygon to split.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::SplitPolygon Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Splits a polygon along a curve.

```
AcMap::EErrCode SplitPolygon(  
    AcMapObjectPtrArray& apResults,  
    const AcDbObjectId& curveId,  
    long lPolygonId  
);
```

Parameters	Description
apResults	Output new and affected topological objects.
curveId	Input curve to split at.
lPolygonId	Input ID of polygon to split.

## Returns

Returns `AcMap::EErrCode kOk` if successful; otherwise, returns a different error code.

## Remarks

After the split succeeds, the polygon object reference with ID `lPolygonId` becomes invalid and the caller is responsible for retrieving the new object reference.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

AcMapTopology::TraceBestPath Method

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the least-cost path between start and end points with intermediate points to visit.

```
AcMap::EErrCode TraceBestPath(  
    AcMapObjectPtrArray& apObjectsOnPath,  
    const AcMapTopoNode& start,  
    const AcMapTopoNode& end,  
    const AcMapNodePtrArray& apIntermediates,  
    const AcMapTraceParameters& params  
);
```

Parameters	Description
apObjectsOnPath	Output nodes and lines along the path.
start	Input AcMapTopoNodenode that starts the path.
end	Input AcMapTopoNodenode that ends the path.
apIntermediates	Input array of intermediate nodes to visit.
params	Input <a href="#">AcMapTraceParameters</a> trace parameter settings.
Returns	

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::TraceFlood Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Traces the flood from a specified point.

```
AcMap::EErrorCode TraceFlood(  
    AcMapObjectPtrArray& apObjectsOnPath,  
    const AcMapTopoNode& start,  
    const AcMapFloodParameters& params  
);
```

Parameters	Description
apObjectsOnPath	Output nodes and lines on the trace flood path.
start	Input AcMapTopoNodenode that starts the trace.
params	Input <a href="#">AcMapFloodParameters</a> flood trace parameter settings.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopology Class](#), [AcMapTopology Class](#)

[AcMapTopology::TraceLeastCostPath Method](#)

[AcMapTopology Class](#) | [AcMapTopology Class](#)

Finds the least-cost path between two points.

```
AcMap::EErrorCode TraceLeastCostPath(  
    AcMapObjectPtrArray& apObjectsOnPath,  
    const AcMapTopoNode& start,  
    const AcMapTopoNode& end,  
    const AcMapTraceParameters& params  
);
```

Parameters	Description
apObjectsOnPath	Output nodes and lines along the path.
start	Input AcMapTopoNodenode that starts the path.
end	Input AcMapTopoNodenode that ends the path.
params	Input <a href="#">AcMapTraceParameters</a> trace parameter settings.
Returns	

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)  
AcMapTopologySource:: ~AcMapTopologySource Destructor  
[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopologySource();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)  
AcMapTopologySource:: AcMapTopologySource Constructor  
[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Constructs an instance of this class.

```
AcMapTopologySource();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)  
[AcMapTopologySource:: AcMapTopologySource Constructor](#)  
[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Constructs an instance of this class by using a copy constructor.

```
AcMapTopologySource(  
    const AcMapTopologySource& src  
);
```

Parameters	Description
src	Input reference to an object of class AcMapTopologySource.

## Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)

[AcMapTopologySource:: GetName Method](#)

[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Retrieves the topology name.

```
const ACHAR* GetName() const;
```

Returns

Returns the topology name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)

[AcMapTopologySource:: GetPath Method](#)

[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Retrieves the full path of the topology.

```
const ACHAR* GetPath() const;
```

Returns

Returns the full path of the topology.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)

[AcMapTopologySource:: GetSource Method](#)

[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Retrieves the source of the topology.

```
const ACHAR* GetSource() const;
```

Returns

Returns the source of the topology.

Remarks

The source string will be "DWG" (the current drawing) or "Attached DWG".

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopologySource Class](#), [AcMapTopologySource Class](#)

[AcMapTopologySource:: IsLoaded Method](#)

[AcMapTopologySource Class](#) | [AcMapTopologySource Class](#)

Determines whether the topology is loaded.

```
bool IsLoaded() const;
```

Returns

Returns true if the topology is loaded, or false if it is not loaded.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: ~AcMapTopoNode Destructor](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoNode();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: GetEdges Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves this node's edges in counter-clockwise order.

```
AcMap::EErrorCode GetEdges(  
    AcMapHalfEdgePtrArray& apEdges  
);
```

Parameters	Description
apEdges	Output array of half edges.
Returns	

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::EErrorCode kErrTopTraceLinkNotExist if no edge exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: GetEntity Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the object identifier of the AutoCAD entity associated with this node.

```
virtual AcMap::EErrorCode GetEntity(  
    AcDbObjectId& featureID  
) const;
```

Parameters	Description
featureID	Output the feature's ID.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: GetID Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the unique identifier of this node in its topology.

```
virtual long GetID() const;
```

Returns

Returns the ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: GetLocation Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the position of this node.

```
AcMap::EErrorCode GetLocation(  
    AcGePoint3d& point  
) const;
```

Parameters	Description
point	Output position.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: GetNextEdge Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the next edge given an incoming full edge and a left or right turn direction.

```
AcMap::EErrCode GetNextEdge(  
    AcMapTopoFullEdge*& pOutGoingEdge,  
    const AcMapTopoFullEdge* pInComingEdge,  
    bool bGoLeft  
);
```

Parameters	Description
pOutGoingEdge	Output next AcMapTopoFullEdgeedge. This value points to a copy of the full edge referenced by the full edge object. The caller is responsible for freeing this object.
pInComingEdge	Input incoming AcMapTopoFullEdgefull edge.
bGoLeft	Input true to turn left, or false to turn right.

Returns

Returns AcMap::EErrCode kOk if successful. Returns AcMap::EErrCode kErrTopTraceLinkNotExist if no edge exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: GetNextEdge Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the next edge given an incoming half edge and a left or right turn direction.

```
AcMap::EErrorCode GetNextEdge(  
    AcMapTopoHalfEdge*& pOutGoingEdge,  
    const AcMapTopoHalfEdge* pInComingEdge,  
    bool bGoLeft  
);
```

Parameters	Description
pOutGoingEdge	Output next AcMapTopoHalfEdgeedge. This value points to a copy of the half edge referenced by the half edge object. The caller is responsible for freeing this object.
pInComingEdge	Input incoming AcMapTopoHalfEdgehalf edge.
bGoLeft	Input true to turn left, or false to turn right.

Returns

Returns AcMap::EErrorCode kOk if successful. Returns AcMap::EErrorCode kErrTopTraceLinkNotExist if no edge exists.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: GetResistance Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the resistance of this node.

```
double GetResistance();
```

Returns

Returns the node resistance.

Remarks

The measurement units depend on your application. The default resistance is zero.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: GetTopology Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Retrieves the topology that contains this node.

```
AcMap::EErrorCode GetTopology(  
    const AcMapTopology*& pTopology  
);
```

Parameters	Description
pTopology	Output AcMapTopologytopology reference. Do not delete this reference.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: IsMoveable Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Determines whether this node is moveable - this function is not yet implemented in AutoCAD Map and always returns true.

```
bool IsMoveable();
```

Returns

Returns true if this node is moveable; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: IsOnThisObject Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Determines whether a specified point is within tolerance of this node.

```
virtual bool IsOnThisObject(  
    const AcGePoint3d& point,  
    double dTolerance  
) const;
```

Parameters	Description
point	Input point to test.
dTolerance	Input maximum distance to the point. The measurement units depend on the current drawing's unit settings.

Returns

Returns true if the point is within tolerance of this node, or false if it is not.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

[AcMapTopoNode:: SetIsMoveable Method](#)

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Sets whether this node is moveable - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrCode SetIsMoveable(  
    bool bMoveable  
);
```

Parameters

Description

bMoveable

Input true to make this node moveable; otherwise, false.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: SetLocation Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Sets the position of this node - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrorCode SetLocation(  
    const AcGePoint3d & point  
);
```

Parameters	Description
point	Input new position.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoNode Class](#), [AcMapTopoNode Class](#)

AcMapTopoNode:: SetResistance Method

[AcMapTopoNode Class](#) | [AcMapTopoNode Class](#)

Sets the resistance of this node.

```
AcMap::EErrorCode SetResistance(  
    double dResistance  
);
```

Parameters	Description
dResistance	Input resistance.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

## Remarks

The measurement units depend on your application. The default resistance is zero.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
AcMapTopoOverlayData:: ~AcMapTopoOverlayData Destructor  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoOverlayData();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
AcMapTopoOverlayData:: AcMapTopoOverlayData Constructor  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Constructs an instance of this class.

```
AcMapTopoOverlayData();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
AcMapTopoOverlayData:: AcMapTopoOverlayData Constructor  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Constructs an instance of this class by using a copy constructor.

```
AcMapTopoOverlayData(  
    const AcMapTopoOverlayData& src  
);
```

Parameters	Description
src	Input reference to an object of class AcMapTopoOverlayData.

## Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)

[AcMapTopoOverlayData:: GetColumnName Method](#)

[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Retrieves the column name of the object data table.

```
const ACHAR* GetColumnName() const;
```

Returns

Returns the column name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
[AcMapTopoOverlayData:: GetColumnType Method](#)  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Retrieves the column data type of the object data table.

```
AcMap::EDataType GetColumnType() const;
```

Returns

Returns the column data type.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)

[AcMapTopoOverlayData:: GetExpression Method](#)

[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Retrieves the expression.

```
const ACHAR* GetExpression() const;
```

Returns

Returns the expression.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoOverlayData Class](#), [AcMapTopoOverlayData Class](#)  
[AcMapTopoOverlayData:: SetColumnType Method](#)  
[AcMapTopoOverlayData Class](#) | [AcMapTopoOverlayData Class](#)

Sets the column data type of the object data table.

```
AcMap::EErrCode SetColumnType(  
    AcMap::EDataType enmColumnType  
);
```

Parameters	Description
enmColumnType	Input column data type.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

[AcMapTopoPolygon:: ~AcMapTopoPolygon Destructor](#)

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoPolygon();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

[AcMapTopoPolygon:: GetArea Method](#)

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the area of this polygon.

**double** GetArea();

Returns

Returns the polygon area.

Remarks

The measurement units depend on the current drawing's unit settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
AcMapTopoPolygon:: GetBoundary Method  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the boundary of this polygon, represented by a list of rings.

```
AcMap::EErrorCode GetBoundary(  
    AcMapRingPtrArray& apBoundingRings  
);
```

Parameters	Description
apBoundingRings	Output array of rings.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
[AcMapTopoPolygon:: GetCentroid Method](#)  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the position of this polygon's centroid.

```
AcMap::EErrCode GetCentroid(  
    AcGePoint3d& centroid  
) const;
```

Parameters	Description
centroid	Output centroid position.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

AcMapTopoPolygon:: GetEntity Method

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the object identifier of the AutoCAD entity associated with this polygon.

```
virtual AcMap::EErrCode GetEntity(  
    AcDbObjectId& featureID  
) const;
```

Parameters	Description
featureID	Output the feature's ID.

Returns

Returns AcMap::EErrCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
AcMapTopoPolygon:: GetHierChildren Method  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the hierarchical child polygons, if any, of this polygon - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrorCode GetHierChildren(  
    AcMapPolygonPtrArray& apChildren  
);
```

Parameters	Description
apChildren	Output array of child polygons.

### Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
AcMapTopoPolygon:: GetHierParent Method  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the hierarchical parent polygon, if any, of this polygon - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrorCode GetHierParent(  
    AcMapTopoPolygon*& pParent  
);
```

Parameters	Description
pParent	Output parent polygon.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

[AcMapTopoPolygon:: GetID Method](#)

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the unique identifier of this polygon in its topology.

```
virtual long GetID() const;
```

Returns

Returns the ID.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

AcMapTopoPolygon:: GetParent Method

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the multi-polygon parent, if any, of this polygon - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrorCode GetParent(  
    AcMapMultiPolygon*& pParent  
);
```

Parameters	Description
pParent	Output multi-polygon parent.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
AcMapTopoPolygon:: GetPerimeter Method  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the perimeter of this polygon.

```
double GetPerimeter();
```

Returns

Returns the polygon perimeter.

Remarks

The measurement units depend on the current drawing's unit settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)  
AcMapTopoPolygon:: GetTopology Method  
[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Retrieves the topology that contains this polygon.

```
AcMap::EErrorCode GetTopology(  
    const AcMapTopology*& pTopology  
);
```

Parameters	Description
pTopology	Output AcMapTopologytopology reference. Do not delete this reference.

## Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

[AcMapTopoPolygon:: IsOnThisObject Method](#)

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Determines whether a specified point is within tolerance of this polygon.

```
virtual bool IsOnThisObject(  
    const AcGePoint3d& point,  
    double dTolerance  
) const;
```

Parameters	Description
point	Input point to test. This point must be inside or on the boundary, regardless of tolerance.
dTolerance	(Ignored) Input maximum distance to the point.
Returns	

Returns true if the point is within tolerance of this polygon, or false if it is not.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoPolygon Class](#), [AcMapTopoPolygon Class](#)

AcMapTopoPolygon:: Traverse Method

[AcMapTopoPolygon Class](#) | [AcMapTopoPolygon Class](#)

Traverses to another polygon by following a curve - this function is not yet implemented in AutoCAD Map.

```
AcMap::EErrorCode Traverse(  
    AcMapTopoPolygon*& pPolygon,  
    const AcDbObjectId& path  
);
```

Parameters	Description
pPolygon	Output destination polygon.
path	Input path curve to follow.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)  
AcMapTopoRing:: ~AcMapTopoRing Destructor  
[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTopoRing();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)

[AcMapTopoRing:: GetArea Method](#)

[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Retrieves the area of this ring.

**double** GetArea();

Returns

Returns the ring area.

Remarks

The measurement units depend on the current drawing's unit settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)

AcMapTopoRing:: GetEdges Method

[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Retrieves the edges that make up this ring.

```
AcMap::EErrorCode GetEdges(  
    AcMapHalfEdgePtrArray& apEdges  
);
```

Parameters	Description
apEdges	Output array of edges.

Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)

AcMapTopoRing:: GetLength Method

[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Retrieves the length of this ring.

**double** GetLength();

Returns

Returns the ring length.

Remarks

The measurement units depend on the current drawing's unit settings.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)

AcMapTopoRing:: GetStartEdge Method

[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Retrieves the first edge in this ring.

```
AcMap::EErrorCode GetStartEdge(  
    AcMapTopoHalfEdge*& pStartEdge  
);
```

### Parameters

### Description

pStartEdge

Output AcMapTopoHalfEdgeedge. This value points to a copy of the half edge referenced by the half edge object. The caller is responsible for freeing this object.

### Returns

Returns AcMap::EErrorCode kOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTopoRing Class](#), [AcMapTopoRing Class](#)

AcMapTopoRing::IsExterior Method

[AcMapTopoRing Class](#) | [AcMapTopoRing Class](#)

Determines whether this ring is exterior (that is, an outer boundary of its polygon).

```
bool IsExterior();
```

Returns

Returns true if this ring is exterior; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapTraceParameters Class](#), [AcMapTraceParameters Class](#)  
AcMapTraceParameters:: ~AcMapTraceParameters Destructor  
[AcMapTraceParameters Class](#) | [AcMapTraceParameters Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTraceParameters();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).



## Drawing Cleanup Functions

### [Topology Function Synopsis](#)

---

The functions for cleaning topology objects before building the topology begin with `tpm_clean`.

<a href="#">tpm_cleanactionlistdel</a>	Deletes a cleanup action from the action list.
<a href="#">tpm_cleanactionlistgetat</a>	Gets the cleanup action at a given list position.
<a href="#">tpm_cleanactionlistins</a>	Inserts a cleanup action in the action list.
<a href="#">tpm_cleanactionlistqty</a>	Gets the number of cleanup actions in the action list.
<a href="#">tpm_cleanalloc</a>	Allocates the cleanup model.
<a href="#">tpm_cleananchors</a>	Gets a cleanup model's anchored entities.
<a href="#">tpm_cleancancel</a>	Cancel the cleanup process without updating the drawing.
<a href="#">tpm_cleancomplete</a>	Tests for a next group to be cleaned.
<a href="#">tpm_cleancreatedss</a>	Gets created entities following a cleanup process.
<a href="#">tpm_cleanend</a>	Completes the cleanup process and updates the drawing.
<a href="#">tpm_cleanerrorcur</a>	Sets the next error to clean in the current group.
<a href="#">tpm_cleanerrordraw</a>	Creates a temporary marker for the current error.
<a href="#">tpm_cleanerrorfix</a>	Fixes the current error.
<a href="#">tpm_cleanerrorget</a>	Gets the coordinates of the current error.
<a href="#">tpm_cleanerrormark</a>	Creates a persistent marker for the current error.
<a href="#">tpm_cleanerrorset</a>	Sets the coordinates of an error fix point.
<a href="#">tpm_cleanfree</a>	Frees the cleanup model.
<a href="#">tpm_cleangroupdraw</a>	Creates temporary markers for all errors of the current group.
<a href="#">tpm_cleangroupfix</a>	Fixes all errors of the current group.

<a href="#">tpm_cleangroupmark</a>	Creates persistent markers for all errors of the current group..
<a href="#">tpm_cleangroupnext</a>	Goes to the next error group.
<a href="#">tpm_cleangroupqty</a>	Counts the errors in the current group.
<a href="#">tpm_cleangroupsubtype</a>	Determines the subtype of the current group.
<a href="#">tpm_cleangrouptype</a>	Determines the type of the current group.
<a href="#">tpm_cleanincludes</a>	Gets a cleanup model's target entities.
<a href="#">tpm_cleaninit</a>	Initializes the cleanup model.
<a href="#">tpm_cleaninitanchorset</a>	Specifies anchored entities for the cleanup model.
<a href="#">tpm_cleanmodifiedss</a>	Gets changed entities following a cleanup process.
<a href="#">tpm_cleanprofileload</a>	Loads a drawing cleanup profile.
<a href="#">tpm_cleanprofilesave</a>	Saves a drawing cleanup profile.
<a href="#">tpm_cleanstart</a>	Starts the cleanup process.
<a href="#">tpm_cleanunchangedss</a>	Gets unchanged entities following a cleanup process.



## Topology Configuration Variables

[Topology Global-Function API](#)

---

Topology configuration variables include three subsets:

<a href="#">Cleanup variables</a>	Properties for cleanup models.
<a href="#">Cleanup action variables</a>	Properties for cleanup actions.
<a href="#">Topology variables</a>	Properties for topologies.

To allocate a set of configuration variables, regardless of which subset you will be using, use [tpm\\_varalloc](#), which returns an `ade_id` for the set that it creates.

To free a set of configuration variables, use [tpm\\_varfree](#).

To get the value of a configuration variable, use [tpm\\_varget](#).

To set the value of a configuration variable, use [tpm\\_varset](#).

To get the properties of a given cleanup action, first allocate a new set of variables, and then use [tpm\\_cleanactionlistgetat](#).

To get the properties of a given topology, first allocate a new set of variables, and then use [tpm\\_infobuildvar](#).

To list all the values of a given set of variables, use [tpm\\_varlist](#).



## tpm\_acclose

[Access Functions](#)

---

Closes a topology.

int

```
tpm_acclose(  
    ade_id tpm_id);
```

Returns **RTNORM** or an error code.

`tpm_id`     The topology ID returned by [tpm\\_acopen](#).

When you close a topology, it remains in memory until you unload it.



## tpm\_acexist

### [Access Functions](#)

---

Checks whether a topology exists.

```
int  
tpm_acexist(  
    char *toponame,  
    int source,  
    int loaded);
```

Returns **RTNORM** if the topology exists, or **RTERROR** if no topology exists with the specified name.

- |                 |  |
|-----------------|--|
| <b>toponame</b> | The topology name for which to check.  |
| <b>source</b>   | The source flag, either <b>1</b> (check topologies in the current and source drawings) or <b>0</b> (check the current drawing only). The default is <b>0</b> . |
| <b>loaded</b>   | The loaded in memory flag, either <b>1</b> (check only topologies in memory) or <b>0</b> (check all topologies). The default is <b>0</b> .                     |



## tpm\_acload

### [Access Functions](#)

---

Loads a topology into memory.

```
int  
tpm_acload(  
    char *toponame,  
    int source);
```

Returns **RTNORM** or an error code.

<b>toponame</b>	The name of the topology to load.
<b>source</b>	The source flag, either <b>1</b> (read from source drawings only) or <b>0</b> (read from current drawing only). The default is <b>0</b> .

Before loading a topology, you must first test for its existence with [tpm\\_acexist](#).

Topology information is stored in the drawing, but the topology is not automatically loaded when you open the drawing. If you want, for example, to query, edit, or overlay a topology, you must load it into memory. Once loaded, the topology remains in memory until you unload it.



## tpm\_acopen

### [Access Functions](#)

---

Opens a topology.

```
ade_id  
tpm_acopen(  
    char *toponame,  
    int write_access);
```

Returns the topology ID or **ADE\_NULLID**.

**toponame**        The name of the topology to open.

**write\_access**    Access status, either **1** (write) or **0** (read only). The default is **0**.

The **tpm\_acopen** function opens a topology and creates a new **tpm\_id** that provides access to it.

If your application opens a topology with write access, your user is the only one who can edit this topology. No other user can even open it. If your application opens topology with read access, other users can open it also, but with read access only. To find out the access status of an already-open topology, use [tpm\\_infostatus](#).

If you use **tpm\_acopen** to test the status of a topology, always close the topology with [tpm\\_acclose](#) to ensure that you do not leave the topology open with multiple IDs pointing to it.

A topology loaded from a source drawing cannot be opened for write access. To find out if a topology was loaded from a current drawing, use [tpm\\_infocurrent](#).

**Important** You must store the topology ID when it is returned by **tpm\_acopen**, and make sure that you do not lose it. If you do not have the ID of an open topology, you have no way to get it, and you cannot close the topology. Your alternatives are to quit AutoCAD Map or start a new drawing.



## tpm\_acqty

[Access Functions](#)

---

Counts the number of topologies.

int

```
tpm_acqty(  
    int source,  
    int loaded);
```

Returns the number of topologies or an error code.

**source** The source flag, either **1** (count topologies in current and source drawings) or **0** (count topologies in current drawing only). The default is **0**.

**loaded** The loaded in memory flag, either **1** (count only topologies in memory) or **0** (count all topologies). The default is **0**.



## tpm\_acunload

[Access Functions](#)

---

Unloads a topology from memory.

```
int  
tpm_acunload(  
    char *toponame);
```

Returns **RTNORM** or an error code.

**toponame**      The name of the topology to unload.

Before a topology can be unloaded, all topology IDs that reference it must be closed with [tpm\\_acclose](#).



## tpm\_acupgradeopen

[Access Functions](#)

---

Changes the access of an open topology from read-only to write.

```
int  
tpm_acupgradeopen(  
    ade_id tpm_id);
```

Returns **RTNORM** or an error code.

**tpm\_id**     The topology ID returned by [tpm\\_acopen](#).

If a topology is already open with write access, you cannot change its access with this function. In addition, you cannot change the read-only access of a topology loaded from a source drawing.



## tpm\_anabuffer

### [Analyzing Functions](#)

---

Creates a buffer around a specified topology and stores the results in a new topology.

```
int  
tpm_anabuffer(  
    ade_id source_id,  
    char *offset,  
    ade_id var_id,  
    char *result_name,  
    char *result_desc);
```

Returns **RTNORM** or an error code.

<code>source_id</code>	The topology ID of the topology you want to buffer.
<code>offset</code>	The buffer offset distance (a standard data extension expression).
<code>var_id</code>	Topology variables ID.
<code>result_name</code>	The name of the resulting buffer topology or <b>NULL</b> if you want to omit this parameter.
<code>result_desc</code>	The description of the resulting buffer topology or <b>NULL</b> if you want to omit this parameter.

The topology variables ID references a set of [topology variables](#).

This function draws one or more buffer perimeters. If the result argument is omitted or **NULL**, the buffered topologies are AutoCAD objects only. Otherwise the function creates a new topology that is loaded but not open.

The source topology must be loaded and open. The Read/Write access of the source topology does not affect this function.

A buffer is a zone that is drawn around a topology using a specified offset value. You can buffer any of the three topology types, node, network, or polygon. For example, you might specify a buffer on either side of a river to show the extent of a flood plain.



## tpm\_anadissolve

### [Analyzing Functions](#)

---

Combines topology elements that have the same value in the specified field and stores the results in a new topology.

Use this function to create a topology that displays less specific geographic information than the original topology. For example, if a data table attached to a map of city blocks has a field that lists the blocks as commercial or residential, a dissolve can create a new topology in which city block boundaries have been removed.

int

```
tpm_anadissolve(  
    ade_id source_id,  
    char *field,  
    ade_id var_id,  
    char *result_name,  
    char *result_desc,  
    char *objTable,  
    char *objColumn);
```

Returns **RTNORM** or an error code.

<b>source_id</b>	The topology ID of the topology you want to dissolve. You can dissolve polygon or network topologies only.
<b>field</b>	The field used to dissolve the topology. The field is a standard data extension expression, for example, <b>:Value@Tab1</b> , <b>.Layer</b> . Topology elements that share the same data value in this field will be combined.
<b>var_id</b>	The topology variables ID.
<b>result_name</b>	The name of the resulting topology, which will be loaded but not open. If argument is omitted or <b>NULL</b> , the dissolve results in a collection of AutoCAD objects.
<b>result_desc</b>	The description of the resulting topology or <b>NULL</b> .
<b>objTable</b>	The name of the object data table in the resulting topology that will store the shared dissolve field value or <b>NULL</b> .
<b>objColumn</b>	The name of the field in the object data table that will store the shared dissolve field

value or **NULL**.

The last four parameters are mutually dependent. If you want to omit these parameters, use **NULL** as a placeholder.

The topology variables ID references a set of [topology variables](#).

The function works for polygon or network topologies. If it finds two or more attached objects that have the same value in the specified dissolve field, it dissolves them.

- Dissolving polygons deletes their centroids and shared links and creates a new polygon with a new centroid.
- Dissolving links deletes shared nodes. The links are merged into one polyline.

The function writes the shared dissolve field value to the object data table and field that you specify and attaches the table to the result object. If this table does not exist, the function creates it. If the table you specify is already attached to one of the source objects, a table with default values is attached to the result object.



## tpm\_anaoverlay

### [Analyzing Functions](#)

---

Overlays two topologies.

```
int
tpm_anaoverlay(
    ade_id overlay_id,
    struct resbuf *overlay_data,
    ade_id source_id,
    struct resbuf *source_data,
    int oper,
    ade_id var_id,
    char *obj-table,
    char *obj_tabledesc,
    char *result_name,
    char *result_desc);
```

Returns **RTNORM** or an error code

<code>overlay_id</code>	Overlay topology ID of topology 1. Must be a polygon topology.
<code>overlay_data</code>	Overlay data expression composed of AutoCAD Map expressions and output object data columns or <b>nil</b> . See the Overlay Data Expressions section below.
<code>source_id</code>	Source topology ID of topology 2. Must be a polygon topology.
<code>source_data</code>	Overlay data expression composed of AutoCAD Map expressions and output object data columns or <b>nil</b> . See the Overlay Data Expressions section below.
<code>oper</code>	Overlay operation. See the Overlay Operations table below.
<code>var_id</code>	ID of topology variables for building the result topology.
<code>obj_table</code>	Name of the new object data table that will contain the final data (created by the function). Cannot be an existing table.
<code>obj_tabledesc</code>	Object data table description.
<code>result_name</code>	Result topology name. Omit if function results produces AutoCAD objects instead of a topology.

<code>result_desc</code>	Result topology description.
<code>overlay_data</code>	Pointer to result buffer of overlay attributes.
<code>source_data</code>	Pointer to result buffer of source attributes.

See the Visual LISP description of [tpm\\_anaoverlay](#) for information about the list elements in the `expr1 resbuf`.

If the `result_name` argument is `NULL`, the result is a collection of AutoCAD objects. Otherwise it is a new topology that is loaded but not open. The last four parameters have the same mutual dependencies in the `ADSRX` function as they do in the Visual LISP equivalent, except that you cannot omit any of them. You must use `NULL`.

The source topology must be of polygon type for union and paste operations. For all other operations, it can be of any topology type, such as node, network, and polygon.

The topology variables ID references a set of [topology variables](#).

## Overlay Data Expressions

The `overlay_data` and `source_data` arguments are `resbufs` that are constructed as follows:

```
struct resbuf *list1;
list1 = acutBuildList(
    RTLB
    RTSTR expr1
    RTSTR colname1
    RTSTR coldsc1
    RTSHORT coltype1
    RTLE
    RTLB
    RTSTR expr2
    ...
    RTLE
    ...
    0);
```

## Overlay Data Expression Arguments

<code>expr1</code>	AutoCAD Map expression ( <code>RTSTR</code> ).
<code>col_name1</code>	Object data column name ( <code>RTSTR</code> ).
<code>col_desc1</code>	Object data column description ( <code>RTSTR</code> ).

`col_type1` Object data column type (**RTSHORT**): 1 through 4, where 1 = integer, 2 = real, 3 = character, and 4 = point.

## Overlay Operations

1	Intersect
2	Union
3	Identity
4	Erase
5	Clip
6	Paste

## Result Topology

If the `result_name` argument is omitted or `nil`, the result is a collection of AutoCAD objects. Otherwise, it is a new topology that is loaded but not open.

You must specify the list of data values to come from each input topology and the specific data for each.

Additionally, you must specify the name of the result object data table in the `objTable` parameter to contain the final data. If you do not specify a result table, no data is attached to the resulting topology elements. The function creates this table. If the result table you specify already exists, the function returns an error and cancels the overlay process.

Data derived by the overlay process is also attached to the result object data table. This data is written for each topology element in the resulting topology. It is written for each polygon if the source is a polygon topology, for each link if it is a network topology, or for each node if it is a node topology. The table always includes the following fields:

### Field Names in Result Object Data Table

<code>TOPO_ID</code>	Element ID of the new element in the result topology
<code>T1_ID</code>	Element ID of the parent polygon in the overlay topology
<code>T1_PERCENTAREA</code>	Area of the new polygon in the result topology compared to the area of the parent polygon in the overlay topology. Written only if both overlay and result are polygon topologies
<code>T2_ID</code>	Element ID of parent element in the source topology. The parent element can be a node, link, or polygon.
<code>T2_PERCENTAREA</code>	Area of the new polygon in the resulting topology compared to the area of the

parent polygon in the source topology. Written only if both source and result are polygon topologies

The result table includes these fields along with the fields that you specify in the arguments you supply for the `overlay_data` and `source_data` parameters. Each field name is prefixed with `T1_` or `T2_` to indicate which topology its data comes from.

For example, if you specify `FIELD1`, `FIELD2` and `FIELD3` from table `SOIL` for the first topology and `FIELD1`, `FIELD4` and `FIELD5` from table `WATER` for the second, the result table has the following fields:

```
T1_SOIL_FIELD1
T1_SOIL_FIELD2
T1_SOIL_FIELD3
T1_WATER_FIELD1
T1_WATER_FIELD4
T1_WATER_FIELD5
```

The following sample overlays two polygon topologies which are opened and loaded using `tpm_acload()` and `tpm_acopen()` respectively. A `resbuf` is created which contains data expressions associated with object data attached to the source topology. `tpm_anaoverlay()` is called with all required parameters. The resulting topology and object data table contains information about the percentage of the source polygons consumed by the overlay, (buffer topology), the median household income and the population density per square mile among others. The input topologies are closed and then unloaded using `tpm_acclose()` and `tpm_acunload()` respectively. The `resbufs` are then released as required.

```
char* pszOverlayTopoName = "I-95_Buffer";
int topoWriteAccess = 0;
int returnCode = tpm_acload(pszOverlayTopoName, 0);
ade_id topoForOverlayId = tpm_acopen(pszOverlayTopoName, topoWriteAccess);
struct resbuf* pOverlayDataRb = NULL;
char* pszSourceTopoName = "blkgrp";
returnCode = tpm_acload(pszSourceTopoName, 0);
ade_id sourceTopoId = tpm_acopen(pszSourceTopoName, topoWriteAccess);
struct resbuf* pSourceDataRb = acutBuildList(
    RTLB,
    RTSTR, ":MEDIAN_VAL@RIBLKGRP", // Data Extension expression
    RTSTR, "MEDIAN_VAL",         // Object data column name
    RTSTR, "",                   // Object data column description
    RTSHORT, 2,                  // Object data column type
    RTLE,
    RTLB,
    RTSTR, ":POP90_SQMI@RIBLKGRP",
    RTSTR, "POP90_SQMI",
```

```

        RTSTR, "",
        RTSHORT, 1,
        RTLE,
        0);
int overlayOperation = 1;
ade_id topoConfigVarId = tpm_varalloc();
char* pszODTable = "BufferOverlayBlkGrp";
char* pszODTableDesc = "BufferOverlayBlkGrp-Desc";
char* pszResultTopoName = "BuffrOverBlkGrp";
char* pszResultTopoDesc = "BuffrOverBlkGrp-Desc";
returnCode = tpm_anaoverlay(
    topoForOverlayId,
    pOverlayDataRb,
    sourceTopoId,
    pSourceDataRb,
    overlayOperation,
    topoConfigVarId,
    pszODTable,
    pszODTableDesc,
    pszResultTopoName,
    pszResultTopoDesc
);
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe overlay operation was successfully completed.");
}
else {
    acutPrintf(
        "\nThe overlay operation could not be performed.");
}
tpm_acclose(topoForOverlayId);
tpm_acunload(pszOverlayTopoName);
tpm_acclose(sourceTopoId);
tpm_acunload(pszSourceTopoName);
acutRelRb(pOverlayDataRb);
acutRelRb(pSourceDataRb);

```



## tpm\_cleanactionlistdel

### [Cleanup Functions](#)

---

Deletes a cleanup action from the action list.

int

```
tpm_cleanactionlistdel  
    ade_id clean_var_id,  
    long index);
```

Returns **RTNORM** or an error code.

**clean\_var\_id**      The cleanup variables ID returned by [tpm\\_varalloc](#).

**index**             The position in the list of the action to delete.

The **clean\_var\_id** argument references properties for the cleanup action that you are preparing to initiate (see [Cleanup Variables](#)). These properties include the action list.

The **index** argument is a zero-based position in the action list. A value greater than or equal to the list size or less than **0** returns an error.



## tpm\_cleanactionlistins

### [Cleanup Functions](#)

---

Inserts a cleanup action in the action list.

```
int  
tpm_cleanactionlistins  
    ade_id clean_var_id,  
    long index,  
    int action,  
    ade_id action_var_id);
```

Returns **RTNORM** or an error code.

<code>clean_var_id</code>	The cleanup variables ID returned by <a href="#">tpm_varalloc</a> .
<code>index</code>	Where to insert in the action list.
<code>action</code>	The cleanup action to insert, a clean group type. See <a href="#">tpm_cleangrouptype</a> for a list of types.
<code>action_var_id</code>	The cleanup action variables ID returned by <a href="#">tpm_varalloc</a> .

The `clean_var_id` argument references properties for the cleanup operation that you are preparing to initiate (see [Cleanup Variables](#)). These properties include the action list.

The `index` argument is a zero-based position in the action list, or `-1` for the last position. A value greater than or equal to the list size or less than `-1` is taken as `-1`.

**Note** When you insert the Simplify Objects action (clean group type 128), it is always listed first, and you cannot insert it more than once.

The `action_var_id` argument references properties affecting the specific cleanup action that you are inserting (see [Cleanup Action Variables](#)). Use [tpm\\_varset](#) to set them before calling `tpm_cleanactionlistins`.



## tpm\_cleanactionlistqty

### [Cleanup Functions](#)

---

Gets the number of cleanup actions in the action list.

```
int  
tpm_cleanactionlistqty  
    ade_id clean_var_id,  
    long *qty);
```

Returns **RTNORM** or an error code.

**clean\_var\_id**      The cleanup variables ID returned by [tpm\\_varalloc](#).

**qty**                Outputs the number of cleanup actions.

The **clean\_var\_id** argument references properties for the cleanup operation that you are preparing to initiate. See [Cleanup Variables](#). These properties include the action list.

The function passes the number of cleanup actions through a parameter (as a **long**) instead of returning it (as a **real**) as the AutoLISP function does.



## tpm\_cleanalloc

### [Cleanup Functions](#)

---

Allocates the cleanup model.

```
ade_id  
tpm_cleanalloc();
```

Returns the model ID (**real**) or **ADE\_NULLID**.

To clean the objects before they become the elements of a topology, you must construct a model of these objects and their relationships. You can use this model to discover and repair drawing errors that would prevent topology creation.

The following example allocates a cleanup model, initializes the cleanup model, counts errors by group type and subtype, and frees the cleanup model.

```
ade_id var_id = ADE_NULLID;    // variable ID  
ade_id cln_id = ADE_NULLID;    // clean ID  
ads_name ss;                  // selection set  
long qty = 0;                 // quantity  
int type = 0;                 // clean error group type  
int subtype = 0;              // clean error group subtype  
int done = 0;  
int result = 0;  
  
var_id = tpm_varalloc();  
cln_id = tpm_cleanalloc();  
  
if( ! var_id || ! cln_id ) {  
    ads_printf("\nMemory allocation failed.");  
    tpm_varfree(var_id);  
    tpm_cleanfree(cln_id);  
  
    return;  
}  
  
// initialize a set of objects, in selection set ss,  
// to clean
```

```

tpm_cleaninit(cln_id, var_id, ss);
result = tpm_cleanstart(cln_id);

if( result != RTNORM ) {
    ads_printf("\nClean startup failed.");
    tpm_varfree(var_id);
    tpm_cleanfree(cln_id);
    return;
}

// Count errors by group type and subtype
while ( ! done) {
result = tpm_cleangroupNext(cln_id);
    if ( result == RTNORM ) {
        result = tpm_cleancomplete(cln_id);
        if ( result == RTNORM )
            done = 1;
        else {
            type = tpm_cleangrouptype(cln_id);
            subtype = tpm_cleangroupsubtype(cln_id);
            tpm_cleangroupqty(cln_id, &qty);
            ads_printf("Group type, subtype: %d,%d
                Number of errors: %d", type, subtype, qty);
        } // else
    } // if
} // while

tpm_cleanfree(cln_id);
tpm_varfree(var_id);

```



## tpm\_cleananchorss

### [Cleanup Functions](#)

---

Gets the anchored entities in a cleanup model.

int

tpm\_cleananchorss

```
ade_id clean_var_id,  
ads_name ss);
```

Returns **RTNORM** or an error code.

**clean\_var\_id**      The cleanup variables ID returned by [tpm\\_varalloc](#).

**ss**                      The anchored entities, a selection set.

Call this function after calling [tpm\\_cleaninit](#).



## tpm\_cleancancel

### [Cleanup Functions](#)

---

Cancels the cleanup process without updating the drawing.

```
int  
tpm_cleancancel(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

**clean\_id**     The cleanup model ID returned by [tpm\\_cleanalloc](#).

The function clears the cleanup model without updating the drawing. Do one of the following:

- Because the model is still allocated, you can call [tpm\\_cleaninit](#) using the same cleanup model ID. You can then use [tpm\\_cleanstart](#) to start the cleanup process.
- Because **tpm\_cleancancel** does not cancel initialized values, you can clear the cleanup model and continue without calling **tpm\_cleaninit**.



## tpm\_cleancomplete

### [Cleanup Functions](#)

---

Tests for a next group to clean.

int

```
tpm_cleancomplete(  
    ade_id clean_id);
```

Returns **TRUE** if no more groups need cleaning, or **FALSE**.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

You must call this function after [tpm\\_cleangroupNext](#) to determine if another group needs cleaning, because [tpm\\_cleangroupNext](#) does not provide this information.

You must usually execute several cleanup loops, because fixing one error sometimes causes others. After you process all the groups in the selection set, call [tpm\\_cleanstart](#) to return to the beginning, and then call [tpm\\_cleancomplete](#) to test if cleaning is complete. Repeat the loop until cleaning is complete.

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).



## tpm\_cleancreatedss

### [Cleanup Functions](#)

---

Gets created entities following a drawing cleanup.

```
int  
tpm_cleancreatedss  
    ads_id clean_id,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

**ads\_id** Cleanup model ID returned by [tpm\\_cleanalloc](#).

**ss** Selection set of created entities.

This function gets entities that were created during the cleanup process.

**Note** Call this function after calling [tpm\\_cleanend](#), which concludes the cleanup process. If you call this function earlier, it returns a selection set from the previous cleanup or the empty selection set.



## tpm\_cleanend

### [Cleanup Functions](#)

---

Completes the cleanup process and updates the drawing.

```
int  
tpm_cleanend(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

The function resets the cleanup model and updates the drawing. It fixes errors marked with the [tpm\\_cleanerrorfix](#) function. The model is still allocated. It is possible to call [tpm\\_cleaninit](#) using the same cleanup model ID.

Each error has some default method, which **tpm\_cleanend** uses during error fixing. To change this method, use [tpm\\_cleanerrorset](#).



## tpm\_cleanerrorcur

### [Cleanup Functions](#)

---

Sets the next error to clean in the current group.

```
int  
tpm_cleanerrorcur(  
    ade_id clean_id,  
    long index);
```

Returns **RTNORM** or an error code.

**clean\_id**      The model ID returned by [tpm\\_cleanalloc](#).

**index**        The index of the error to clean. The index of the first error is **0**.

The next error to clean is also called the current error.



## tpm\_cleanerrordraw

### [Cleanup Functions](#)

---

Creates a temporary marker for the current error.

int

```
tpm_cleanerrordraw(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

Unlike the persistent markers drawn by [tpm\\_cleanerrormark](#), these markers are deleted when the drawing redraws.



## tpm\_cleanerrorfix

### [Cleanup Functions](#)

---

Fixes the current error.

```
int  
tpm_cleanerrorfix(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

This function marks the current error to fix. It fixes the errors in the cleanup model, but does not fix the drawing. The objects in the drawing are not fixed until you call [tpm\\_cleanend](#).

Each error has a default method that [tpm\\_cleanend](#) uses during error fixing. You can use [tpm\\_cleanerrorset](#) to change this method. For example, the default method for the dangling node error is to erase the link. If you call [tpm\\_cleanerrorset](#) for this error, the dangling node is moved to a new position, but is not erased.



## tpm\_cleanerrorget

### [Cleanup Functions](#)

---

Gets the coordinates of the current error point.

int

```
tpm_cleanerrorget(  
    ade_id clean_id,  
    ads_point coords);
```

Returns **RTNORM** or an error code.

**clean\_id**      The model ID returned by [tpm\\_cleanalloc](#).

**coords**        The coordinates of the error point.

The error point is a misplaced node. For example, for a line undershoot, the end of the unattached line is the error point.

To specify the error to get, use [tpm\\_cleanerrorcur](#).



## tpm\_cleanerrormark

### [Cleanup Functions](#)

---

Creates a persistent marker for the current error.

int

```
tpm_cleanerrormark(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

Unlike the temporary marker drawn by [tpm\\_cleanerrordraw](#), these markers are AutoCAD objects, and they become part of the drawing until you perform another cleanup. AutoCAD Map automatically erases persistent markers from any previous cleanup process.



## tpm\_cleanerrorset

### [Cleanup Functions](#)

---

Sets the coordinates of an error fix point.

```
int  
tpm_cleanerrorset(  
    ade_id clean_id,  
    ads_point coords);
```

Returns **RTNORM** or an error code.

**clean\_id**      The model ID returned by [tpm\\_cleanalloc](#).

**coords**        2D point.

You can use this function to change the default method used by [tpm\\_cleanend](#) during error fixing. For example, the default method for the dangling node error is to erase the link. If you call **tpm\_cleanerrorset** for this error, the dangling node is moved to the new position, but is not erased.

You can also use this function to merge a cluster of points to the point you specify.

To tell the clean engine which error to set, use [tpm\\_cleanerrorcur](#).



## tpm\_cleanfree

### [Cleanup Functions](#)

---

Frees the cleanup model.

int

```
tpm_cleanfree(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

For information about preparing the cleanup model, see [Drawing Cleanup](#).

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).



## tpm\_cleangroupdraw

### [Cleanup Functions](#)

---

Creates temporary markers for all errors in the current group.

int

```
tpm_cleangroupdraw(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

*clean\_id*     The model ID returned by [tpm\\_cleanalloc](#).



## tpm\_cleangroupfix

### [Cleanup Functions](#)

---

Fixes all errors in the current group.

```
int  
tpm_cleangroupfix(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

This function marks errors in the entire current group for fixing. It fixes the errors in the cleanup model, but does not fix the drawing. The objects in the drawing are not fixed until you call [tpm\\_cleanend](#).

To mark only the current error, use [tpm\\_cleanerrorfix](#).



## tpm\_cleangroupmark

### [Cleanup Functions](#)

---

Creates persistent markers for all errors in the current group.

int

```
tpm_cleangroupmark(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

*clean\_id*     The model ID returned by [tpm\\_cleanalloc](#).



## tpm\_cleangroupnext

[Cleanup Functions](#)

---

Goes to the next error group.

```
int  
tpm_cleangroupnext(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

The function always returns **RTNORM**, and fails only when the `clean_id` parameter is invalid. It cannot inform you when there are no more groups to clean. To check for this condition, use [tpm\\_cleancomplete](#).

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).



## tpm\_cleangroupqty

### [Cleanup Functions](#)

---

Counts the errors in the current group.

```
int  
tpm_cleangroupqty(  
    ade_id clean_id,  
    long *qty);
```

Returns **RTNORM** or an error code.

*clean\_id*      The model ID returned by [tpm\\_cleanalloc](#).

*qty*            The error count (**long**).

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).



## tpm\_cleangroupsubtype

### [Cleanup Functions](#)

---

Determines the subtype of the current group.

```
int  
tpm_cleangroupsubtype(  
    ade_id clean_id);
```

Returns a subtype code or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).

### Subtype Codes

1	Degenerate entities (subtype of Erase Short Objects)
2	Short entities (subtype of Erase Short Objects)
3	Short segments (subtype of Erase Short Objects)
4	Vertex is near to segment (subtype of Extend Undershoots)
5	Vertex is near to vertex (subtype of Extend Undershoots)

Codes 1 through 3 are subtypes of the short type. Codes 4 and 5 are subtypes of the undershoot type. This function is useful for short and undershoot errors only.

**Degenerate Entities** A degenerate polyline has only one vertex. This invalid vertex type sometimes results from the drawing cleanup process, and can be removed by further drawing cleanup.



## tpm\_cleangrouptype

### [Cleanup Functions](#)

---

Determines the type of the current group.

```
int  
tpm_cleangrouptype(  
    ade_id clean_id);
```

Returns a type code or an error code.

`clean_id`     The model ID returned by [tpm\\_cleanalloc](#).

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).

### Type Codes

1	Erase short objects
2	Break crossing objects
4	Extend undershoots
8	Delete duplicates
16	Snap clustered nodes
32	Dissolve pseudo nodes
64	Erase dangling objects
128	Simplify objects
256	Zero length objects
512	Apparent intersections
1024	Weed polylines



## tpm\_cleanincluess

### [Cleanup Functions](#)

---

Gets a cleanup model's Include set.

```
int  
tpm_cleanincluess  
    ade_id clean_var_id,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

`clean_var_id`      The cleanup variables ID returned by [tpm\\_varalloc](#).

`ss`                The entities to be cleaned, a selection set.

Call this function after calling [tpm\\_cleaninit](#).



## tpm\_cleaninit

### [Cleanup Functions](#)

---

Initializes the cleanup model.

```
int  
tpm_cleaninit(  
    ade_id clean_id,  
    ade_id var_id,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

- clean\_id**     The model ID returned by [tpm\\_cleanalloc](#).
- var\_id**       The cleanup variables ID, which references a set of [cleanup variables](#). This is the ID that is returned by [tpm\\_varalloc](#).
- ss**            The selection set or **NULL**. See note below about **INCLUDEOBS\_AUTOSELECT**.

You can free the selection set after the cleanup model has been initialized.

For more information about the cleanup model, see [Drawing Cleanup](#).

The function reads the cleanup settings and the selected entities into the cleanup model.

If the cleanup variable **INCLUDEOBS\_AUTOSELECT** is set to **1** (select all), all entities in the drawing will be included, regardless of the **ss** argument, in which case this argument can be **NULL** instead of a selection set, as the following code sample illustrates.

If the cleanup variable **INCLUDEOBS\_LAYERS** is set to **"\*"** (all layers), all **ss** entities will be included. If it contains a list of layers, **ss** entities will be included only if they reside on those layers.

For an example that shows how you can use the cleanup functions, see [tpm\\_cleanalloc](#).

The following sample shows outlines the steps leading up to the use of [tpm\\_cleaninit\(\)](#). Allocate memory for the cleanup model using [tpm\\_cleanalloc\(\)](#), then allocate a set of configuration variables using [tpm\\_varalloc\(\)](#). Define the cleanup variable "INCLUDEOBS\_AUTOSELECT", (which must be used with a filtered selection set), then call [tpm\\_varset\(\)](#) to set it. Define a selection set of objects to clean and finally, combine the cleanup variables and the selection set of objects using [tpm\\_cleaninit\(\)](#). Release any **resbufs** as required and free the selection set.

---

```
ade_id cleanupModelId = tpm_cleanalloc();
ade_id cleanupVarId = tpm_varalloc();
char* pszConfigVarName = "INCLUDEOBS_AUTOSELECT";
struct resbuf* pIncludeObjsVarValRb = acutBuildList(
    RTSHORT, 0,
    0);
int resultCode = tpm_varset(
    cleanupVarId,
    pszConfigVarName,
    pIncludeObjsVarValRb);

acutRelRb(pIncludeObjsVarValRb);

ads_name ssObjsForCleanup;
acedSSGet("X", NULL, NULL, NULL, ssObjsForCleanup);

resultCode = tpm_cleaninit(
    cleanupModelId,
    cleanupVarId,
    ssObjsForCleanup);

resultCode = acedSSFree(ssObjsForCleanup);
```



## tpm\_cleaninitanchorset

### [Cleanup Functions](#)

---

Specifies anchored entities for the cleanup model.

int

```
tpm_cleaninitanchorset  
    ade_id clean_id,  
    ade_id clean_var_id,  
    ads_name ssAnchor);
```

Returns **RTNORM** or an error code.

<code>clean_id</code>	The cleanup model ID returned by <a href="#">tpm_cleanalloc</a> .
<code>clean_var_id</code>	The cleanup variables ID returned by <a href="#">tpm_varalloc</a> .
<code>ssanchor</code>	The entities to be anchored (a selection set).

Anchored entities remain fixed in position during the cleanup process. You can free the selection set after anchored entities have been specified.

Call `tpm_cleaninitanchorset` before calling [tpm\\_cleaninit](#).

The `clean_var_id` argument references properties for the cleanup operation that you are preparing to initiate (see [Cleanup Variables](#)).

If the cleanup variable `ANCHOROBJES_LAYERS` is set to "\*" (all layers), all `ssAnchor` entities will be anchored. If it contains a list of layers, `ssAnchor` entities will be anchored only if they reside on those layers.



## tpm\_cleanmodifiedss

### [Cleanup Functions](#)

---

Gets changed entities following a drawing cleanup.

```
int  
tpm_tpm_cleanmodifiedss  
    ads_id clean_id,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

**ads\_id** Cleanup model ID returned by [tpm\\_cleanalloc](#).

**ss** Selection set of changed entities.

This function gets members of the [Include](#) set that were changed during the cleanup process.

**Note** Call this function after calling [tpm\\_cleanend](#), which concludes the cleanup process. If you call this function earlier, it returns a selection set from the previous cleanup or the empty selection set.



## tpm\_cleanprofileload

### [Cleanup Functions](#)

---

Loads a drawing cleanup profile (.dpf file).

```
int  
tpm_cleanprofileload  
    ade_id clean_var_id,  
    char *filename);
```

Returns **RTNORM** or an error code.

**clean\_var\_id**      The cleanup variables ID returned by [tpm\\_varalloc](#).

**filename**          The full path and .dpf file name.

Loading a profile will reset all properties referenced by the **clean\_var\_id** argument.



## tpm\_cleanprofilesave

### [Cleanup Functions](#)

---

Saves a drawing cleanup profile (.dpf file).

int

```
tpm_cleanprofilesave  
    ade_id clean_var_id,  
    char *filename);
```

Returns **RTNORM** or an error code.

`clean_var_id`      The cleanup variables ID returned by [tpm\\_varalloc](#).

`filename`          The full path and .dpf file name.



## tpm\_cleanstart

### [Cleanup Functions](#)

---

Starts the cleanup process.

```
int  
tpm_cleanstart(  
    ade_id clean_id);
```

Returns **RTNORM** or an error code.

`clean_id`     The cleanup model ID returned by [tpm\\_cleanalloc](#).

The function finds the first group type or subtype that contains errors. See [tpm\\_cleangrouptype](#) and [tpm\\_cleangroupsubtype](#) for lists of types.

Cleanup is an iterative process. Depending on cleanup options and processing order, you may need to run the cleanup loop several times to achieve the desired results. After selecting the groups to process with **tpm\_cleangrouptype** and processing all the groups in the selection set, call [tpm\\_cleancomplete](#) to test if cleaning is complete. If complete, call **tpm\_cleanstart** and repeat the cleanup loop.

For an example of the cleanup process, see [tpm\\_cleanalloc](#).



## tpm\_cleanunchangedss

### [Cleanup Functions](#)

---

Gets unchanged entities following a drawing cleanup.

```
int  
tpm_cleanunchangedss  
    ads_id clean_id,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

**ads\_id** Cleanup model ID returned by [tpm\\_cleanalloc](#).  
**ss** Selection set of unchanged entities.

This function gets members of the [Include](#) set that were not changed during the cleanup process. Note that it does not get members of the [Anchor](#) set, which are unchanged by definition.

**Note** Call this function after calling [tpm\\_cleanend](#), which concludes the cleanup process. If you call this function earlier, it returns a selection set from the previous cleanup or the empty selection set.



## tpm\_editaddelem

### [Editing Functions](#)

---

Adds an element to a topology.

```
int  
tpm_editaddelem(  
    ade_id tpm_id,  
    int type,  
    struct resbuf *res);
```

Returns **RTNORM** or an error code.

- tpm\_id**     The topology ID.
- type**        The element's type code: **1**, **2**, or **3**, where **1** = Node, **2** = Link, and **3** = Polygon.
- res**         The element to add. Depending on the **type** argument, specify one of the following: If **1**, specify a point or the entity name of a point object. If **2**, specify the entity name of a line object. If **3**, specify a selection set.

The topology must be open with Write access, (1). If you add a node to a link, the link is split.

Set up the **resbuf** for the **res** argument like this:

Resbuf Element	Value
<b>restype</b>	Depends on the function's <b>type</b> argument: <b>1</b> , <b>2</b> , or <b>3</b> . If <b>1</b> , use <b>RTPOINT</b> , <b>RT3DPOINT</b> (if the topology's <b>CREATE_NODE</b> variable is set), or <b>RTENAME</b> . If <b>2</b> , use <b>RTENAME</b> . If <b>3</b> , use <b>RTPICKS</b> .
<b>resval</b>	Depending on the given <b>restype</b> , a new point, the entity name of an existing line or point entity, or a selection set. If the topology's <b>CREATE_NODE</b> variable is set (see <a href="#">tpm_varset</a> ), and you specify a point that does not yet exist, it is created and added to the topology.
<b>rbnext</b>	<b>NULL</b> (that is, no more data).

For a polygon topology only: If you add a link that closes an open sequence of links or is connected to the

topology at both ends of the link, `tpm_editaddelem` creates a new polygon.

The following sample adds a new element to an existing network topology. A `resbuf` is created containing the information required to draw a new pline. The pline is drawn using `acedCmd()`. The new pline's entity id is then obtained using `acdbEntLast()`. The topology that will receive the new link is loaded and opened using `tpm_acload()` and `tpm_acopen()` respectively. A `resbuf` containing the new pline entity is constructed, then `tpm_editaddelem()` is called with all required parameters. Appropriate status messages are displayed based on the return code. Topologies are closed and unloaded from memory and the `resbufs` are released as required.

```
struct resbuf* pNewPlineRb = acutBuildList(
    RTSTR, "_pline",
    RTSTR, "54.0816,50.8243",
    RTSTR, "68,53",
    RTSTR, "",
    0);
int returnCode = acedCmd(pNewPlineRb);
ads_name newPline;
returnCode = acdbEntLast(newPline);
char* pszTopoName = "NetTopo";
int topoWriteAccess = 1;
returnCode = tpm_acload(pszTopoName, 0);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
struct resbuf* pNewElementRb = acutBuildList(RTENAME, newPline, 0);
int elementType = 2;
returnCode = tpm_editaddelem(topoId, elementType, pNewElementRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe new element was successfully added.");
}
else {
    acutPrintf(
        "\nThe new element was not added.");
}
tpm_acclose(topoId);
tpm_acunload(pszTopoName);
acutRelRb(pNewPlineRb);
acutRelRb(pNewElementRb);
```



## tpm\_editdelelem

### [Editing Functions](#)

---

Deletes an element from a topology.

```
int
tpm_editdelelem(
    ade_id tpm_id,
    ade_id elem_id,
    int delobj);
```

Returns **RTNORM** or an error code.

<code>tpm_id</code>	The topology ID.
<code>elem_id</code>	The ID of the desired element.
<code>delobj</code>	Delete status, either 0 (deletes the object from the topology only) or non-zero (deletes the object from the drawing also).

The following example deletes from memory and erases all the links of a network topology and counts the number of successful deletions.

```
void Delete_links()
{
    int indx = 0;    // Element index
    int dindx = 0;  // Deleted element count
    long qty = 0;
    ade_id tpm_id = ADE_NULLID;
    ade_id elm_id = ADE_NULLID;
    int result = 0;

    // load or build a network topology here
    // and also use tpm_acopen to assign a tpm_id for the topology

    // get the number of links
    tpm_elemqty (tpm_id, 2, &qty);
```

```

// delete all of the topology links

for ( ; indx < qty ; ) {
    // get the links ID
    elm_id = tpm_lemid(tpm_id, 2, indx);
    if ( elm_id == ADE_NULLID ) {
        indx++;
        ads_printf("\nUnable to obtain an element ID.");
    }
    else {
        // delete element and erase object
        result = tpm_editdelelem(tpm_id, elm_id, 1);

        if ( result == RTNORM )
            dindx++;
        else
            indx++;
    }
}

} // delete_links

```

### Notes and Warnings

- This function does not erase corresponding entities in the drawing unless the **delobj** argument is set to something other than **0**.
- Deleting an element can cause other deletions.
  - If you delete a node, you delete any link or polygon that contains it.
  - If you delete a link, you delete only the nodes belonging to that link. If the link belongs to one polygon only, you delete the polygon. If the link is shared by two polygons, you merge the polygons.
  - If you delete a polygon, you delete any node or link belonging to that polygon only.



## tpm\_editmodelem

### [Editing Functions](#)

---

Modifies a topology element.

```
int
tpm_editmodelem(
    ade_id tpm_id,
    ade_id elem_id,
    struct resbuf *new_val);
```

Returns **RTNORM** or an error code.

<b>tpm_id</b>	The topology ID
<b>elem_id</b>	The ID of the desired element.
<b>new_val</b>	A list consisting of a code for the property to modify and a new value for the property. See Properties and Values below.

The following code constructs a **resbuf** for a **new\_val** point argument:

```
ads_point newpoint;
newpoint[X] = 1.0;
newpoint[Y] = 1.0;
newpoint[Z] = 0.0;

struct resbuf* new_val = ads_buildlist (
    RTLB,
    RTSHORT, 10,
    RTPOINT, newpoint,
    RTLE,
    0);
```

**Note** The dotted pair is not supported for point data.

The following examples of build **resbuf** data structures for the **new\_val** argument.

```
// For a point value, given a point, p
```

```

rb = acutBuildList(
    RTLB, RTSHORT, 10, RTPPOINT, p, RTLE, 0);
// For an integer value
rb = acutBuildList(
    RTLB, RTSHORT, code, RTSHORT, value, RTDOTE, 0);
// For a real value
rb = acutBuildList(
    RTLB, RTSHORT, code, RTREAL, value, RTDOTE, 0);

```

You must release the **resbuf**.

## Properties and Values

(10 . point)	New coordinates of node or centroid ( <b>RTPPOINT</b> )
(40 . f_res)	Resistance of node ( <b>RTREAL</b> ), or forward resistance of link
(41 . r_res)	Reverse resistance of link ( <b>RTREAL</b> )
(70 . dir)	Link direction ( <b>RTSHORT</b> ): -1 Reverse 0 Bidirectional 1 Forward

The following example modifies the direction variable on a link element in the "MyNetworkTopo" topology. The topology is first loaded into memory using **tpm\_acload()**, then opened for write using **tpm\_acopen()**. A **resbuf** is constructed containing the property/value pair associated with link direction. After calling **tpm\_editmodelem()** with all required parameters, the **returnCode** is evaluated and an appropriate message is displayed. The **resbuf** is then released as required.

```

char* pszTopoName = "NetTopo";
int topoWriteAccess = 1;
int returnCode = tpm_acload(pszTopoName, NULL);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
int elementType = 2;
long elementIndex = 1;
ade_id elementID = tpm_lemid(topoId, elementType, elementIndex);
struct resbuf* pModElementRb = acutBuildList(
    RTLB,
    RTSHORT, 70,
    RTSHORT, -1,
    RTDOTE,
    0);
returnCode = tpm_editmodelem(topoId, elementID, pModElementRb);

```

```
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe element modification was successful.");
}
else {
    acutPrintf(
        "\nThe element modification was not successful.");
}
acutRelRb(pModElementRb);
tpm_acclose(topoId);
tpm_acunload(pszTopoName);
```



## tpm\_editupdelem

### [Editing Functions](#)

---

Updates a topology element.

int

```
tpm_editupdelem(  
    ade_id tpm_id,  
    ade_id elem_id);
```

Returns **RTNORM** if the object was successfully updated or an error code if it was not.

*tpm\_id*      The topology ID.

*elem\_id*     The ID of the desired element.

This function updates a topology element so that it reflects the current state of the corresponding entity in AutoCAD. This function is normally used in conjunction with AutoCAD Notification so that changes made in AutoCAD can be reflected in the topology model.



## tpm\_elemadj

### [Element Information Functions](#)

---

Compiles a list of adjacent elements for the specified element.

```
struct resbuf
*tpm_elemadj(
    ade_id tpm_id,
    ade_id elem_id,
    int adj_type);
```

Returns a list of element IDs or **NULL**.

**tpm\_id**        The topology ID.

**elem\_id**        An element ID, received from [tpm\\_elemid](#).

**adj\_type**        The type of adjacent elements to be compiled into the list. Values can be: **1** (node), **2** (link), or **3** (polygon).

The following sample populates a **resbuf** with adjacent elements, (links) on a network topology using [tpm\\_elemadj\(\)](#). If the operation is successful the adjacent id(s) are displayed, and the **resbuf** is released as required. Note, [tpm\\_acopen\(\)](#) was used to obtain the loaded topology id and [tpm\\_elemid\(\)](#) was used to obtain a specific element id within the loaded topology.

```
char* pszTopoName = "NetTopo";
int topoWriteAccess = 0;
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
int adjElementType = 2;
long elementIndex = 1;
ade_id adjElementID = tpm_elemid(topoId, adjElementType, elementIndex);
struct resbuf* pAdjElementIdsRb = tpm_elemadj(topoId, adjElementID, adjElementType);
if (NULL != pAdjElementIdsRb){
    struct resbuf* rb = pAdjElementIdsRb;
    acutPrintf(
        "\nThe specified element has an ID of %.0lf. The following adjacent ID(s) were detected:"
        , adjElementID);
    while(NULL != rb) {
```

```
    acutPrintf(
        "\n\t %.0lf"
        , rb->resval.rreal);
    rb = rb->rbnext;
}
}
else {
    acutPrintf(
        "\nNo adjacent elements were detected.");
}
acutRelRb(pAdjElementIdsRb);
tpm_acclose(topoId);
```



## tpm\_elemfind

### [Element Information Functions](#)

---

Finds an element within a topology.

```
ade_id  
tpm_elemfind(  
    ade_id tpm_id,  
    int type,  
    struct resbuf *pattern);
```

Returns an element ID or **ADE\_NULLID**.

- tpm\_id** The topology ID.
- type** The type of element to find. Values can be: **1** (node), **2** (link), or **3** (polygon).
- pattern** The name of the point or entity. If pattern is a point, in which case type must be **1**, the function returns the nearest point or link, or the enclosing polygon. If pattern is an entity name, in which case type can have any value, the function returns the corresponding object.

The following sample finds a specific element in an existing network topology and uses the `tpm_editaddelem` sample as a baseline.

A **resbuf** is created containing the information required to draw a new pline. The pline is drawn using `acedCmd()`. The new pline's entity id is then obtained using `acdbEntLast()`. The topology that will receive the new link is loaded and opened using `tpm_acload()` and `tpm_acopen()` respectively. A **resbuf** containing the new pline entity is constructed, then `tpm_editaddelem()` is called with all required parameters. Appropriate status messages are displayed based on the return code.

To find the new element; A **resbuf** is created containing the **pattern** parameter information, (entity name). `Tpm_elemfind()` is then called with all required parameters. The return value is validated against **ADE\_NULLID** and appropriate status messages are displayed. Topologies are closed and unloaded from memory and the **resbufs** are released as required.

```
struct resbuf* pNewPlineRb = acutBuildList(  
    RTSTR, "_pline",  
    RTSTR, "54.0816,50.8243",  
    RTSTR, "68,53",
```

```

        RTSTR, "",
        0);
int returnCode = acedCmd(pNewPlineRb);
ads_name newPline;
returnCode = acdbEntLast(newPline);
char* pszTopoName = "NetTopo";
int topoWriteAccess = 1;
returnCode = tpm_acload(pszTopoName, 0);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
struct resbuf* pNewElementRb = acutBuildList(RTENAME, newPline, 0);
int elementType = 2;
returnCode = tpm_editaddelem(topoId, elementType, pNewElementRb);
if (RTNORM == returnCode){
    acutPrintf(
        "\nThe new element was successfully added.");
}
else {
    acutPrintf(
        "\nThe new element was not added.");
}

// Find the new element.
struct resbuf* pSeekPatternRb = acutBuildList(RTENAME, newPline, 0);
ade_id elementId = tpm_elemfind(topoId, elementType, pSeekPatternRb);
if (ADE_NULLID != elementId){
    acutPrintf(
        "\nThe new element has a id of: %.0lf"
        , elementId);
}
else {
    acutPrintf(
        "\nThe specified element was not found.");
}
tpm_acclose(topoId);
tpm_acunload(pszTopoName);
acutRelRb(pNewPlineRb);
acutRelRb(pNewElementRb);
acutRelRb(pSeekPatternRb);

```



## tpm\_elemget

### [Element Information Functions](#)

---

Lists information about an element in a topology.

```
struct resbuf
*tpm_elemget(
    ade_id tpm_id,
    ade_id elem_id);
```

Returns a **resbuf** list of value pairs or **NULL**.

**tpm\_id**      The topology ID.  
**elem\_id**     An element ID returned by [tpm\\_elemid](#).

For each value pair in the list that is returned, the first value is an integer code for the information type, and the second value is the information. The list format depends on the element type: node, link, or polygon.

The following examples illustrate possible contents of **resbuf** data structures returned by **tpm\_elemget**.

If the referenced element is a node, the contents of the **resbuf** could be as follows.

Restype	Resval
RTLB	
RTSHORT	0
RTREAL	10.000000
RTDOTE	
RTLB	
RTSHORT	-2
RTENAME	2546033006, 3172571566
RTDOTE	

RTLB	
RTSHORT	-1
RTSHORT	1
RTDOTE	
RT3DPOINT	10.000000, 4.000000, 4.000000
RTLB	
RTSHORT	40
RTREAL	0.000000
RTDOTE	

If the refernced element is a link, the contents of the **resbuf** could be as follows.

Restype	Resval
RTLB	
RTSHORT	0
RTREAL	29.000000
RTDOTE	
RTLB	
RTSHORT	-2
RTENAME	2546045206, 3172507766
RTDOTE	
RTLB	
RTSHORT	-1
RTSHORT	2
RTDOTE	
RTLB	
RTSHORT	1

RTREAL	4.000000
RTDOTE	
RTLB	
RTSHORT	2
RTREAL	8.000000
RTDOTE	
RTLB	
RTSHORT	3
RTREAL	19.000000
RTDOTE	
RTLB	
RTSHORT	4
RTREAL	20.000000
RTDOTE	
RTLB	
RTSHORT	40
RTREAL	2.320675
RTDOTE	
RTLB	
RTSHORT	41
RTREAL	2.320675
RTDOTE	
RTLB	
RTSHORT	70
RTSHORT	0
RTDOTE	

If the referenced element is a polygon, the contents of the **resbuf** could be as follows.

Restype	Resval
RTLB	
RTSHORT	0
RTREAL	20.000000
RTDOT	
RTLB	
RTSHORT	-2
RTENAME	2546047206, 3172505766
RTDOT	
RTLB	
RTSHORT	-1
RTSHORT	3
RTDOT	
RT3DPOINT	10.000000,3.319400,4.589200
RTLB	
RTSHORT	50
RTREAL	12.775065
RTDOT	
RTLB	
RTSHORT	51
RTREAL	11.674836
RTDOT	

For each dotted pair in the list that is returned, the first value is an integer code for the information type, and the second value is the information. The list format depends on the element type: node, link, or polygon.

#### Information List Format for Nodes

Type	Information
0	Persistent topology ID ( <b>RTREAL</b> ).
-1	Element type code ( <b>RTSHORT</b> ). With node lists, always <b>1</b> , meaning <i>node element</i> .
-2	Entity name of the node object ( <b>RTREAL</b> ).
10	Coordinates of the node object ( <b>RTPOINT</b> ).
40	Node resistance ( <b>RTREAL</b> ). Relevant only for nodes belonging to network or polygon topologies.

### Information List Format for Links

Type	Information
0	Persistent topology ID ( <b>RTREAL</b> ).
-1	Element type ( <b>RTSHORT</b> ). With link lists, always <b>2</b> , meaning <i>link element</i> .
-2	Entity name of the link object ( <b>RTREAL</b> ).
1	Topology ID of start node ( <b>RTREAL</b> ).
2	Topology ID of end node ( <b>RTREAL</b> ).
3	Topology ID of left polygon ( <b>RTREAL</b> ). Relevant only if the link belongs to a polygon topology. Links in a polygon topology can belong to two adjacent polygons, one on the left, and one on the right.
4	Topology ID of right polygon that shares this link ( <b>RTREAL</b> ). Relevant only if the link belongs to a polygon topology.
40	Forward resistance of the link ( <b>RTREAL</b> ).
41	Reverse resistance of the link ( <b>RTREAL</b> ).
70	Link direction ( <b>RTSHORT</b> ): <b>-1</b> , <b>0</b> , or <b>1</b> , where <b>-1</b> = Reverse, <b>0</b> = Bidirectional, and <b>1</b> = Forward.

### Information List Format for Polygons

Type	Information
0	Persistent topology ID ( <b>RTREAL</b> ).
-1	Element type ( <b>RTSHORT</b> ). With polygon lists, always <b>3</b> , meaning <i>polygon element</i> .

-2	Entity name of the polygon centroid ( <b>RTREAL</b> ).
10	Coordinates of the polygon centroid ( <b>RTPOINT</b> ).
50	Perimeter of the polygon ( <b>RTREAL</b> ).
51	Area of the polygon ( <b>RTREAL</b> ).

When a topology is built, it is given a set of [object data fields](#). Their purpose is to contain the information listed in the preceding tables.

The following sample loads and then opens a network topology for read using **tpm\_acload()** and **tpm\_acopen()** respectively. A element id is obtained using **tpm\_elemid()** with the topology id returned by **tpm\_acopen()**. **Tpm\_elemget()** is called with all required parameters. A **resbuf** is populated with information associated with the specified element, a portion of which is then displayed. The **resbuf** is then released as required.

```
char* pszTopoName = "NetTopo";
int returnCode = tpm_acload(pszTopoName, NULL);
int topoWriteAccess = 0;
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
int elementType = 2;
long elementIndex = 1;
ade_id elementID = tpm_elemid(topoId, elementType, elementIndex);
struct resbuf* pElementInfoRb = tpm_elemget(topoId, elementID);
if (NULL != pElementInfoRb){
    struct resbuf* rb = pElementInfoRb;
    while(NULL != rb) {
        if (rb->restype == RTSHORT && (rb->resval.rint == 0)) {
            acutPrintf(
                "\nThe specified elements topology id is: %.0lf"
                , rb->rbnext->resval.rreal);
            break;
        }
        rb = rb->rbnext;
    }
}
else {
    acutPrintf(
        "\nThe specified element was not found.");
}
acutRelRb(pElementInfoRb);
tpm_acclose(topoId);
tpm_acunload(pszTopoName);
```





## tpm\_elemid

### [Element Information Functions](#)

---

Gets the ID of an element.

```
ade_id  
tpm_elemid(  
    ade_id tpm_id,  
    int type,  
    long index);
```

Returns an `ade_id` or `ADE_NULLID`.

<code>tpm_id</code>	The topology ID.
<code>type</code>	An element type. Values can be: <b>1</b> (node), <b>2</b> (link), or <b>3</b> (polygon).
<code>index</code>	Element index. The index of the first element is <b>0</b>

The `ade_id` of a topology element is used by the functions [tpm\\_elemget](#), [tpm\\_elemadj](#), [tpm\\_editdeelem](#), and [tpm\\_editmodelem](#).



## tpm\_elemqty

[Element Information Functions](#)

---

Counts topology elements.

```
int  
tpm_elemqty(  
    ade_id tpm_id,  
    int type,  
    long *qty);
```

Returns **RTNORM** or an error code.

tpm_id	Topology ID
type	Element type. Values can be: <b>1</b> (node), <b>2</b> (link), or <b>3</b> (polygon).
qty	A pointer to the element count.



## tpm\_elemss

### [Element Information Functions](#)

---

Creates a selection set of elements of a given type.

```
int  
tpm_elemss(  
    ade_id tpm_id,  
    int type,  
    ads_name ss);
```

Returns **RTNORM** or an error code.

<code>tpm_id</code>	Topology ID.
<code>type</code>	Element type. Values can be: <b>1</b> (node), <b>2</b> (link), or <b>3</b> (polygon).
<code>ss</code>	Selection set ID.



## tpm\_infocomplete

[Topology Information Functions](#)

---

Tests if a polygon topology is complete.

int

```
tpm_infocomplete(  
    ade_id tpm_id);
```

Returns **TRUE** if the topology is complete or **FALSE**.

**tpm\_id**     The topology ID of a polygon topology.

Use this function to verify that a polygon topology is complete before performing an element trace, overlay, or other topology operation. This function applies only to polygon topologies.

A topology is considered complete if all necessary entities exist in the current drawing. It is considered incomplete if at least one polygon is incomplete.

A query can bring an incomplete topology into the current drawing. Because this part has the same name as the complete topology that it came from, the part could be mistaken for the whole. Although you can execute any topology operation on an incomplete topology, the result may not be what you intend.



## tpm\_infocorrect

### [Topology Information Functions](#)

---

Tests whether a topology is correct and performs a geometrical audit.

int

```
tpm_infocorrect(  
    ade_id tpm_id);
```

Returns **TRUE** or **FALSE**.

**tpm\_id**     The topology ID.

If topology entities are changed using drawing tools, the topology information is modified according to the geometrical changes. If this is impossible, the topology becomes incorrect. Use this function to verify that a topology is correct before performing an element trace, overlay, or other topology operation. See also [tpm\\_infomodified](#).

To fix an incorrect topology, try unloading and reloading it. If it does not reload, you must use topology edit to fix the topology. If this does not work, rebuild the topology.

**Note** This function indicates when an error occurs, but does not show its location.



## tpm\_infocurrent

[Topology Information Functions](#)

---

Checks the source from which the topology was loaded.

int

```
tpm_infocurrent(  
    ade_id tpm_id);
```

Returns **1** if the topology is loaded from the current drawing, or **0** if it comes from source drawings.

`tpm_id`    Topology ID.



## tpm\_infodesc

[Topology Information Functions](#)

---

Gets a topology description.

```
int  
tpm_infodesc(  
    ade_id tpm_id,  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code.

<code>tpm_id</code>	Topology ID.
<code>buffer</code>	Pointer to result buffer.
<code>buflen</code>	Result buffer length.



## tpm\_infomodified

[Topology Information Functions](#)

---

Checks if topology elements have been modified using drawing tools.

int

```
tpm_infomodified(  
    ade_id tpm_id);
```

Returns **1** if any elements have been modified, otherwise **NULL**.

**tpm\_id**     The topology ID.

If topology elements have been modified using drawing tools, then possibly they are no longer correct topologies. See also [tpm\\_infocorrect](#).



## tpm\_infoname

[Topology Information Functions](#)

---

Gets a topology name.

```
int  
tpm_infoname(  
    ade_id tpm_id,  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code.

<code>tpm_id</code>	The topology ID.
<code>buffer</code>	Pointer to result buffer.
<code>buflen</code>	Result buffer length.



## tpm\_infostatus

[Topology Information Functions](#)

---

Checks whether the topology is open for Read or Write access.

int

```
tpm_infostatus(  
    ade_id tpm_id);
```

Returns **1** if the topology is open for Write or **0** if open for Read.

`tpm_id`    The topology ID.



## tpm\_infotype

[Topology Information Functions](#)

---

Gets a topology type.

int

```
tpm_infotype(  
    ade_id tpm_id);
```

Returns a topology type code (1 = node, 2 = network, or 3 = polygon), or `ADE_NULLID`.

`tpm_id`     The topology ID.



## tpm\_infoversion

### [Topology Information Functions](#)

---

Gets the version of a topology.

```
int  
tpm_infoversion(  
    ade_id tpm_id,  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code.

<code>tpm_id</code>	The topology ID.
<code>buffer</code>	Pointer to result buffer.
<code>buflen</code>	Result buffer length.

A topology's version is the Data Extension version (ADE) in which the topology was created. For example, "2.026". This function gets the same result as [tpm\\_iterversion](#). The difference is that **tpm\_infoversion** requires a topology ID, and it works only on topologies that are open.



## tpm\_iterdesc

[Topology Iterating Functions](#)

---

Gets the description of a topology.

```
int
tpm_iterdesc(
    ade_id iter_id,
    char *buffer,
    int buflen);
```

Returns **RTNORM** or an error code.

**iter\_id**     Iterator ID.  
**buffer**     Pointer to result buffer.  
**buflen**     Result buffer length.

The function gets a description of the topology that the iterator points to.

For an example that shows how you can use Topology Iterating functions to find all the topologies the system knows about, see [tpm\\_iterstart](#).



## tpm\_itername

[Topology Iterating Functions](#)

---

Gets the name of a topology.

```
int  
tpm_itername(  
    ade_id iter_id,  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code.

**iter\_id**     Iterator ID.  
**buffer**     Buffer for result (**char**).  
**buflen**     Length of buffer (**int**).

The function gets the name of the topology that the iterator points to.

For an example that shows how you can use Topology Iterating functions to find all the topologies the system knows about, see [tpm\\_iterstart](#).



## tpm\_iternext

[Topology Iterating Functions](#)

---

Moves the iterator to the next topology.

```
int  
tpm_iternext(  
    ade_id iter_id);
```

Returns **RTNORM** or, if an error occurs or another topology is not present, **RTERROR**.

**iter\_id**     Iterator ID.

The first time this function is called after [tpm\\_iterstart](#), it sets the iterator on the first topology definition.

For an example that shows how you can use Topology Iterating functions to find all the topologies the system knows about, see [tpm\\_iterstart](#).



## tpm\_iterstart

### [Topology Iterating Functions](#)

---

Allocates a topology iterator.

```
ade_id  
tpm_iterstart(  
    int source,  
    int loaded);
```

Returns an iterator ID (*ade\_id*) or *ADE\_NULLID*.

**source** Source flag. Values can be: **1** (iterate through the current and source drawings), or **0** (iterate through the current drawing only). The default is **0**.

**loaded** Loaded in memory flag. Values can be: **1** (iterate through topologies in memory only), or **0** (iterate through all topologies). The default is **0**.

This function allocates an iterator and positions it before the first topology definition. This behavior has implications to remember when you use the function.

- Because *tpm\_iterstart* always generates an iterator ID, even if the drawing has no topologies to iterate through, the function fails only when it is out of memory.
- Because *tpm\_iterstart* positions the iterator before the first topology definition, the function cannot indicate whether any topologies exist in the drawing. The only way to determine whether the drawing has topologies is to call [tpm\\_internext](#), which fails if no topology exists beyond the current position of the iterator.

You can have more than one iterator running at the same time.

The following sample shows how you can use Topology Iterating functions to find and then describe all the topologies in the current project.

```
// Iterate through the current drawing only (default)  
int nTopoInCurrentAndDwgSet = 0;  
// Iterate through all topologies (default)  
int nTopoInMemory = 0;  
ade_id topoIteratorId = tpm_iterstart(  
    nTopoInCurrentAndDwgSet,
```

```

        nTopoInMemory);
int resultCode = RTERROR;
char* pszTopoName = "";
const int TOPONAMEMAXLEN = 17;
if (ADE_NULLID == topoIteratorId) {
    acutPrintf(
        "\nA topology iterator could not be generated.");
}
else {
    resultCode = tpm_iternext(topoIteratorId);
    if (RTNORM == resultCode) {
        acutPrintf(
            "\nThe following named topologies were found in the current project:");
        do {
            resultCode = tpm_itername(
                topoIteratorId,
                pszTopoName,
                TOPONAMEMAXLEN);
            int nTopoType = tpm_itype(topoIteratorId);
            switch(nTopoType) {
                case 1 :{
                    acutPrintf("\n\n\"%s\" is a node topology.", pszTopoName);
                }
                break;
                case 2 :{
                    acutPrintf("\n\n\"%s\" is a network topology.", pszTopoName);
                }
                break;
                case 3 :{
                    acutPrintf("\n\n\"%s\" is a polygon topology.", pszTopoName);
                }
                break;
                default:{
                    acutPrintf("\nThe topology type can not be determined.");
                }
            }
        } while (RTNORM == (resultCode = tpm_iternext(topoIteratorId)));
    }
    else {
        acutPrintf("\nNo topologies were detected.");
    }
}
}

```



## tpm\_iterstop

[Topology Iterating Functions](#)

---

Frees an iterator.

int

```
tpm_iterstop(  
    ade_id iter_id);
```

Returns **RTNORM** or an error code.

*iter\_id*     Iterator ID.



## tpm\_ityertype

[Topology Iterating Functions](#)

---

Gets the type of the topology.

```
int  
tpm_ityertype(  
    ade_id iter_id);
```

Returns a topology type code (`int`) or `ADE_NULLID`.

`iter_id`     Iterator ID.

The function gets the type of the topology that the iterator is pointing to. The topology type codes are `1` = node, `2` = network, `3` = polygon.

For an example that shows how you can use Topology Iterating functions to find all the topologies the system knows about, see [tpm\\_iterstart](#).



## tpm\_iterversion

[Topology Iterating Functions](#)

---

Gets the version of a topology.

```
int  
tpm_iterversion(  
    ade_id iter_id,  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code.

**iter\_id**     Iterator ID.  
**buffer**     Buffer for result (**char**).  
**buflen**     Length of buffer (**int**).

The function gets the version of the topology that the iterator is pointing to. The version of a topology is the version of ADE in which it was created, for example, "2.026". The function gets the same result as [tpm\\_infoversion](#). The difference is that [tpm\\_infoversion](#) requires a topology ID, and so it works only on topologies that are open.



## tpm\_mntbuild

### [Building and Erasing Functions](#)

---

Builds a topology.

```
int
tpm_mntbuild(
    ade_id var_id,
    char *name,
    char *desc,
    int type,
    ads_name node_ss,
    ads_name link_ss,
    ads_name cntr_ss);
```

Returns **RTNORM** or an error code.

<code>var_id</code>	Topology variables ID.
<code>name</code>	Topology name.
<code>desc</code>	Topology description.
<code>type</code>	Topology type code. Values can be <b>1</b> (node), <b>2</b> (network), or <b>3</b> (polygon).
<code>node_ss</code>	Entity selection set of nodes or <b>NULL</b> .
<code>link_ss</code>	Entity selection set of links or <b>NULL</b> . Omit for node topology.
<code>cntr_ss</code>	Entity selection set of centroids or <b>NULL</b> . Omit for node and network topology.

If you are building a node topology, the `link_ss` and `cntr_ss` arguments should be **NULL**. If you are building a network topology, the `cntr_ss` argument should be **NULL**.

The topology variables ID references a set of [topology variables](#).

The new topology is loaded, but closed. You must open it with [tpm\\_acopen](#).

The following AutoCAD object types are acceptable for topology elements:

- For links: line, arc, circle, 2D and 3D polyline

- For nodes and centroids: point, insert, and text

A node topology can contain only nodes. A network topology can contain nodes or links, but not centroids. A polygon topology can contain all three.

When the topology is built, all links are assigned a default direction of bidirectional (a value of 0). The forward and reverse resistance values are the length of the link. Nodes are assigned a resistance of 0.

When a topology is built, it is given a set of [object data fields](#). These fields contain information about the elements of the topology.



## tpm\_mnterase

### [Building and Erasing Functions](#)

---

Erases a closed topology from the current drawing.

```
int  
tpm_mnterase(  
    char *toponame);
```

Returns **RTNORM** or an error code.

**toponame**      The name of the topology.

This function can erase a closed topology whether or not it is loaded.



## tpm\_mntrebuild

### [Building and Erasing Functions](#)

---

Rebuilds a topology.

```
int  
tpm_mntrebuild(  
    char *toponame);
```

Returns **RTNORM** or an error code.

**toponame**     The name of the topology.

The rebuilt topology is loaded, but closed. You must open it with [tpm\\_acopen](#).

Rebuilding a topology restores all its object data fields to their default values. Any object data fields modified after the topology was built are lost. See [Topology Object Data](#).

Whether you need to rebuild a topology after you change it depends upon the functions used to make the changes.

- If changes were made using AutoCAD drawing and editing functions, you may need to rebuild the topology.  
  
If the AutoCAD alterations introduce an error, the rebuild could fail. If this happens, you must clean the objects again and use [tpm\\_mntbuild](#). You can use [tpm\\_infocorrect](#) to check for errors before attempting [tpm\\_mntrebuild](#).
- If changes were made using Topology functions, you do not have to rebuild the topology. This applies to objects altered with functions such as [tpm\\_editaddelem](#), [tpm\\_editdelelem](#), and [tpm\\_editmodelem](#).



## tpm\_mntrename

[Building and Erasing Functions](#)

---

Renames a topology.

```
int  
tpm_mntrename(  
    char *toponame,  
    char *newname,  
    char *newdesc);
```

Returns **RTNORM** or an error code.

<b>toponame</b>	The name of the topology you wish to rename.
<b>newname</b>	The new name of the topology.
<b>newdesc</b>	The new description of the topology.



## tpm\_qrygetresdesc

### [Topology Query Functions](#)

---

Gets the description of the query result topology.

```
int  
tpm_qrygetresdesc(  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code if there is no result topology or it has no description.

**buffer**      Buffer for result description string (**char**).

**buflen**      Length of buffer for string (**int**).

This function passes the topology description through a parameter.

For the **buffer** parameter, you must allocate at least the number of characters specified by the **buflen** parameter.



## tpm\_qrygetrestopo

### [Topology Query Functions](#)

---

Gets the name of the query result topology.

```
int  
tpm_qrygetrestopo(  
    char *buffer,  
    int buflen);
```

Returns **RTNORM** or an error code if there is no result topology.

**buffer**      Buffer for resulting topology name string (**char**).

**buflen**      Length of buffer for string (**int**).

This function passes the topology name through a parameter.



## tpm\_qrygettoponame

### [Topology Query Functions](#)

---

Gets the name of the query source topology.

```
int  
tpm_qrygettoponame(  
    char *toponame,  
    int buflen);
```

Returns **RTNORM** or an error code if there is no query source topology.

**toponame**      Pointer to topology name (**char**).

**buflen**        Length of buffer for topology name string (**int**).

This function passes the topology name through a parameter.



## tpm\_qrysetrestopo

### [Topology Query Functions](#)

---

Defines or undefines a query result topology.

```
int  
tpm_qrysetrestopo(  
    char *result_name,  
    char *result_desc);
```

Returns **RTNORM** or an error code.

**result\_name**     The name of the query result topology or **NULL**.

**result\_desc**     The description of the query result topology or **NULL**

A description is optional. Its presence or absence has no effect on the query. Regarding the name:

- If no name is specified, then no query result topology is defined. If a definition already exists, it is canceled.
- If the name begins with an asterisk (\*), the next query result is a temporary topology.
- If the name begins with any other acceptable character, the next query result is a standard topology.



## tpm\_qrysettoponame

### [Topology Query Functions](#)

---

Defines or undefines a topology query.

int

```
tpm_qrysettoponame(  
    char *toponame);
```

Returns **RTNORM** or an error code.

**toponame**     The name of the topology to query, or **NULL**.

This function determines whether the current query is a topology query or a standard data extension query as follows:

- If the **toponame** argument is **NULL**, the current query becomes a standard data extension query.
- If the **toponame** argument is a name, the current query becomes a topology query. But if you specify a topology that does not exist, a subsequent call to **ade\_qryexecute** returns **NULL**.

If you specify a topology that does not exist, a subsequent call to **ade\_qryexecute** returns an error code.



## tpm\_tracealloc

### [Network Tracing Functions](#)

---

Allocates the tracing model.

```
ade_id
tpm_tracealloc(
  ade_id tpm_id,
  char *node_res,
  char *link_dir,
  char *link_forward_res,
  char *link_reverse_res);
```

Returns a tracing model ID or **ADE\_NULLID**.

<code>tpm_id</code>	The topology ID.
<code>node_res</code>	Expression for node resistance. <b>NULL</b> = Default resistance.
<code>link_dir</code>	Expression for link direction. <b>NULL</b> = Default direction.
<code>link_forward_res</code>	Expression for forward link resistance. <b>NULL</b> = Default resistance.
<code>link_reverse_res</code>	Expression for reverse link resistance. <b>NULL</b> = Default resistance.

This function sets the values for the specified topology to the parameters you enter. The values stored in [topology object data](#) when the topology was created are used as defaults if you omit parameters. You can enter any valid expression that evaluates to a numeric result.

The tracing model can be used only with a network or polygon topology.

When you enter a value other than **NULL** for any of the optional resistance arguments, this value is used for all objects in the topology of the appropriate type. It overrides the corresponding value attached to the object. For example, if `node_res` is set to **NULL**, the tracing model uses the value attached to topology object data when calculating the trace. If `node_res` is set to 10.0, all nodes in the topology are overridden with the value of 10.0 when the trace is calculated.

If you want to omit the optional parameters, you can either enter **NULL** to invoke a default value or leave out the parameter altogether, as with other Visual LISP functions. However, before you omit optional parameters, note the dependency relationships indicated by bracketed groups. For example, if you want to

use the `link_dir` parameter, you must enter a `node_res` parameter.

**Important!** When using the Topology API to perform a network trace, such as a Best Route analysis, the source topology used to create the trace topology should not be unloaded or erased until after all API calls relating to the trace have been made. This is because the trace topology references the nodes and links in the source topology. It does not create its own. So, in order to do anything with the elements of the trace, the source topology must remain loaded.

For example, the following Best Route code will silently fail:

```
char* pszTopoName = "NetTopo";
int topoWriteAccess = 1;
int returnCode = tpm_acload(pszTopoName, NULL);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);

ade_id networkTraceId = tpm_tracealloc(topoId, NULL, NULL, NULL, NULL);

struct resbuf* pNodeListRb = acutBuildList(
    RTREAL, 7.0,
    RTREAL, 1.0,
    RTREAL, 4.0,
    RTREAL, 9.0,
    RTREAL, 5.0,
    RTREAL, 6.0,
    RTREAL, 8.0,
    0);
ade_id bestRouteTraceId = tpm_tracebestroute(topoId, networkTraceId, pNodeListRb);

acutRelRb(pNodeListRb);
tpm_acclose(topoId);
tpm_acunload(pszTopoName);

ade_id BestRouteTraceElement = tpm_tracebestroutescan(networkTraceId, 3);
```

The simple solution is to defer the calls to `tpm_acclose` and `tpm_acunload` until after all `tpm_tracebestroutescan` calls.



## tpm\_tracebestroute

### [Network Tracing Functions](#)

---

Calculates the best round-trip route.

```
ade_id
tpm_tracebestroute
    ade_id tpm_id,
    ade_id trace_id,
    struct resbuf *nodes);
```

Returns a topology ID or **ADE\_NULLID**.

- tpm\_id**      The topology ID representing the network you are analyzing.
- trace\_id**    The tracing model ID returned by [tpm\\_tracealloc](#).
- nodes**        List of nodes to visit.

The list of nodes is implemented as a **resbuf** chain, which you can create like this:

```
struct resbuf* pNodeListRb = acutBuildList(
    RTREAL, 8.0, // start node
    RTREAL, 1.0, // visit point 1
    RTREAL, 5.0,
    RTREAL, 11.0,
    RTREAL, 6.0,
    RTREAL, 7.0,
    RTREAL, 10.0,
    RTREAL, 13.0,
    RTREAL, 12.0,
    RTREAL, 9.0,
    RTREAL, 4.0,
    RTREAL, 2.0, // visit point 11
    0);
```

The best route topology, whose ID this function returns if successful, is assigned an arbitrary name and is open for read. To get its name, use [tpm\\_infoname](#). To change its name, use [tpm\\_mntrename](#).

For the best route trace to succeed, the total calculated resistance cannot be greater than the value set for the maximum resistance or less than the value set for the minimum resistance. See [tpm\\_tracesetmaxres](#) and [tpm\\_tracesetminres](#). The accumulated resistance value is the total resistance of the nodes and links that make up the best route.

The following sample creates a best route using `tpm_tracebestroute()`. The `tpm_id` param is obtained using `tpm_acload()` and `tpm_acopen()`. The `trace_id` param is obtained using `tpm_tracealloc()`. A `resbuf` containing the list of nodes which serve as visit points is then created. `Tpm_tracebestroute()` is called with all required parameters, the returned value is checked against `ADE_NULLID`. A successful operation displays the name of the best route topology. The `resbuf` is then released as required.

```
char* pszTopoName = "NetTopo";
int topoWriteAccess = 1;
int returnCode = tpm_acload(pszTopoName, NULL);
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
ade_id traceId = tpm_tracealloc(topoId, NULL, NULL, NULL, NULL);
struct resbuf* pNodeListRb = acutBuildList(
    RTREAL, 8.0,
    RTREAL, 1.0,
    RTREAL, 5.0,
    RTREAL, 11.0,
    RTREAL, 6.0,
    RTREAL, 7.0,
    RTREAL, 10.0,
    RTREAL, 13.0,
    RTREAL, 12.0,
    RTREAL, 9.0,
    RTREAL, 4.0,
    RTREAL, 2.0,
    0);
ade_id bestRouteTraceId = tpm_tracebestroute(topoId, traceId, pNodeListRb);
if (bestRouteTraceId != ADE_NULLID) {
    char pszBestRouteName[25];
    int topoNameLen = 25;
    int returnCode = tpm_infoname(bestRouteTraceId, pszBestRouteName, topoNameLen);
    if (RTNORM == returnCode){
        acutPrintf(
            "\n\nThe best route is named: \"%s\"."
            , pszBestRouteName);
    }
    else {
        acutPrintf(
            "\n\nNo best route could be calculated.");
    }
}
```

```
    }  
  }  
  acutRelRb(pNodeListRb);  
  tpm_acclose(topoId);  
  tpm_acunload(pszTopoName);
```



## tpm\_tracebestroutescan

### [Network Tracing Functions](#)

---

Gets the element ID of a link or node in the best route.

```
ade_id  
tpm_tracebestroutescan  
    ade_id trace_id,  
    int flag);
```

Returns a element ID or ADE\_NULLID.

**trace\_id**     The tracing model ID returned by [tpm\\_tracealloc](#).

**flag**         Path element code. Values can be:

- 0 = Current element
- 1 = First element
- 2 = Last element
- 3 = Next element
- 4 = Previous element

First use [tpm\\_tracebestroute](#) to calculate the best route.



## tpm\_tracebestrouteval

### [Network Tracing Functions](#)

---

Calculates the resistance of the best route.

```
int  
tpm_tracebestrouteval  
    ade_id trace_id,  
    ads_real *resist);
```

Returns **RTNORM** or an error code.

`trace_id`      Tracing model ID returned by [tpm\\_tracealloc](#).

`resist`        Resistance.

First use [tpm\\_tracebestroute](#) to calculate the best route.

This function passes the resistance of the best route through a parameter.



## tpm\_traceelemedit

### [Network Tracing Functions](#)

---

Modifies a tracing model element.

```
int
tpm_traceelemedit(
    ade_id trace_id,
    ade_id elem_id,
    struct resbuf *new_val);
```

Returns **RTNORM** or an error code.

**trace\_id** Model ID (returned by [tpm\\_tracealloc](#)).

**elem\_id** Element ID.

**new\_val** List consisting of a code for the property to modify and a new value for the property. See Properties and Values below.

Examples of building a **resbuf** for the **new\_val** argument.

```
// For an integer value
rb = acutBuildList(
    RTLB, RTSHORT, code, RTSHORT, value, RTDOTE, 0);
// For a real value
rb = acutBuildList(
    RTLB, RTSHORT, code, RTREAL, value, RTDOTE, 0);
```

You must release the **resbuf** when you are finished with it.

### Properties and Values

(40 . f_res)	Resistance of node ( <b>RTREAL</b> ), or forward resistance of link
(41 . r_res)	Reverse resistance of link ( <b>RTREAL</b> )
(70 . dir)	Link direction ( <b>RTSHORT</b> ): -1 Reverse

0 Bidirectional  
1 Forward

The following sample uses `tpm_traceelemedit()` to modify an element created using `tpm_tracebestroute()`. The `trace_id` param is obtained using `tpm_tracealloc()` and `elem_id` is obtained using `tpm_elemid()`. A `resbuf` is created which contains the property/value pair required to modify the forward resistance of the specified element, (link). `Tpm_traceelemedit()` is called with all required parameters, and the returned value is checked against `RTNORM`. A message indicating that the element has been modified is displayed, then the `resbuf` is released as required. Note, this example is based upon a trace which has been created, is loaded and open.

```
char* pszTopoName = "BestRouteTopology1";
int topoWriteAccess = 1;
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
ade_id traceId = tpm_tracealloc(topoId, NULL, NULL, NULL, NULL);
int elementType = 2;
long elementIndex = 5;
ade_id elementID = tpm_elemid(topoId, elementType, elementIndex);
struct resbuf* pModElementRb = acutBuildList(
    RTLB,
    RTSHORT, 40,
    RTREAL, 100.0,
    RTDOTE,
    0);
int returnCode = tpm_traceelemedit(traceId, elementID, pModElementRb);
if (RTNORM == returnCode) {
    acutPrintf(
        "\nThe specified element has been modified.");
}
else {
    acutPrintf(
        "\nThe specified element was not modified.");
}
acutRelRb(pModElementRb);
tpm_acclose(topoId);
```



## tpm\_traceelemget

### [Network Tracing Functions](#)

---

Lists information about a tracing model element.

```
struct resbuf  
*tpm_traceelemget(  
    ade_id trace_id,  
    ade_id elem_id);
```

Returns an information list in a **resbuf** or **NULL**

**trace\_id**      Tracing model ID returned by [tpm\\_tracealloc](#)

**elem\_id**      Trace element ID

The contents of a **resbuf** for a node element could be as follows.

Restype	Resval
RTLB	
RTSHORT	-1
RTSHORT	1
RTDOT	
RTLB	
RTSHORT	40
RTREAL	0.000000
RTDOT	

The contents of a **resbuf** for a link element could be as follows.

Restype	Resval
---------	--------

RTLB	
RTSHORT	-1
RTSHORT	2
RTDOTE	
RTLB	
RTSHORT	1
RTREAL	9.000000
RTDOTE	
RTLB	
RTSHORT	2
RTREAL	18.000000
RTDOTE	
RTLB	
RTSHORT	40
RTREAL	3.000000
RTDOTE	
RTLB	
RTSHORT	41
RTREAL	3.000000
RTDOTE	
RTLB	
RTSHORT	70
RTSHORT	0
RTDOTE	

The list format depends on the element type. For each a-list, the first component is an integer code for the information type, and the second is the information.

## List Format for Nodes

(-1 . elem_code)	Element type code ( <b>RTSHORT</b> ). With node lists, always 1, meaning node element.
(40 . resistance)	Node resistance ( <b>RTREAL</b> ).

## List Format for Links

(-1 . elem_code)	Element type code ( <b>RTSHORT</b> ). With link lists, always 2, meaning link element.
( 1 . topo_id)	Topology ID of start node ( <b>RTREAL</b> ).
( 2 . topo_id)	Topology ID of end node ( <b>RTREAL</b> ).
(40 . fwd_resist)	Forward resistance ( <b>RTREAL</b> ).
(41 . rev_resist)	Reverse resistance ( <b>RTREAL</b> ).
(70 . link_dir)	Link direction ( <b>RTSHORT</b> ): -1, 0, or 1. -1 = Reverse 0 = Bidirectional 1 = Forward

The following sample opens a network trace topology for read using `tpm_acopen()`. A tracing model is allocated using `tpm_tracealloc()` with the topology id returned by `tpm_acopen()`. A trace element id is obtained using `tpm_lemid()`, which is used in the call to `tpm_traceelemget()`. A `resbuf` is populated with information associated with the specified element, a portion of which is then displayed. The `resbuf` is then released as required.

```
char* pszTopoName = "BestRouteTopology1";
int topoWriteAccess = 0;
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);
ade_id traceId = tpm_tracealloc(topoId, NULL, NULL, NULL, NULL);
int elementType = 2;
long elementIndex = 5;
ade_id elementID = tpm_lemid(topoId, elementType, elementIndex);
struct resbuf* pTraceElementInfoRb = tpm_traceelemget(traceId, elementID);
if (NULL != pTraceElementInfoRb) {
    struct resbuf* rb = pTraceElementInfoRb;
    while (NULL != rb) {
        if (rb->restype == RTSHORT) {
            const int nTraceVal = rb->resval.rint;
            switch (nTraceVal)
```

```
{
  case 1:
    acutPrintf(
      "\nThe topology id of the starting node is: %0.lf"
      , rb->rbnext->resval.rreal);
    break;
  case 70:
    acutPrintf(
      "\nThe element direction is: %d"
      , rb->rbnext->resval.rint);
    break;
  default:
    break;
}
}
rb = rb->rbnext;
}
}
else {
  acutPrintf(
    "\nThe specified element was not found.");
}
acutRelRb(pTraceElementInfoRb);
tpm_acclose(topoId);
```



## tpm\_traceelemid

### [Network Tracing Functions](#)

---

Returns the ID of the selected element.

```
ade_id  
tpm_traceelemid(  
    ade_id trace_id,  
    int type,  
    long index);
```

Returns an element ID or **NULL**.

**trace\_id**      Tracing model ID (real) returned by [tpm\\_tracealloc](#)

**type**            Element type:  
                  **1** Node  
                  **2** Link

**index**          Element index. The first element's index is **0**

This function returns the ID of the element selected. Use this function to find the trace result in the source topology after performing a trace with [tpm\\_traceshort](#) or [tpm\\_traceflood](#).

The following sample opens a topology, performs a short path trace and gets the ID of the third link of the result path using [tpm\\_traceelemid\(\)](#). The topology Id for the corresponding link is displayed and the topology is unloaded.

```
char* pszTopoName = "NetTopo";  
int topoWriteAccess = 0;  
int resultCode = tpm_acload(pszTopoName, NULL);  
ade_id topoId = tpm_acopen(pszTopoName, topoWriteAccess);  
ade_id traceId = tpm_tracealloc(topoId, NULL, NULL, NULL, NULL);  
ade_id startNode = 8.0;  
ade_id endNode = 13.0;  
resultCode = tpm_traceshort(  
    traceId,  
    startNode,  
    endNode);
```

```
int elementType = 2;// link
long lLinkQuantity = 0;
resultCode = tpm_traceqty(
    traceId,
    elementType,
    &lLinkQuantity);

long lLinkIndex = 2;
ade_id linkId = tpm_traceelemid(
    traceId,
    elementType,
    lLinkIndex);
if (ADE_NULLID != linkId) {
    acutPrintf("\nThe specified link at index, (%d) contained the topology Id of %.0lf."
        , lLinkIndex, linkId);
}
else {
    acutPrintf("\nThe specified link was not found.");
}
resultCode = tpm_acclose(topoId);
resultCode = tpm_acunload(pszTopoName);
```



## tpm\_traceflood

[Network Tracing Functions](#)

---

Calculates flood paths.

```
int  
tpm_traceflood(  
    ade_id trace_id,  
    ade_id start,  
    ads_real maxres);
```

Returns **RTNORM** or an error code.

**trace\_id**     The tracing model ID, returned by [tpm\\_tracealloc](#).  
**start**        The element ID of the start node.  
**maxres**       The maximum allowed accumulated path resistance.

The accumulated resistance value is the total resistance of the nodes and links that make up the flood trace.



## tpm\_tracefree

[Network Tracing Functions](#)

---

Frees a tracing model.

int

```
tpm_tracefree(  
    ade_id trace_id);
```

Returns **RTNORM** or an error code.

**trace\_id**     The tracing model ID returned by [tpm\\_tracealloc](#).



## tpm\_traceqty

### [Network Tracing Functions](#)

---

Counts the selected elements after a trace.

```
int  
tpm_traceqty(  
    ade_id trace_id,  
    int type,  
    long *qty);
```

Returns **RTNORM** or an error code.

**trace\_id**     The tracing model ID, returned by [tpm\\_tracealloc](#).  
**type**         Element type, either **1** (node) or **2** (link).  
**qty**           Element count.



## tpm\_tracesetmaxres

### [Network Tracing Functions](#)

---

Sets maximum resistance for the shortest path algorithm.

int

```
tpm_tracesetmaxres(  
    ade_id trace_id,  
    ads_real maxres);
```

Returns **RTNORM** or an error code.

**trace\_id**      The tracing model ID, returned from [tpm\\_tracealloc](#).

**maxres**        The maximum allowed accumulated path resistance.

This function sets an accumulated resistance value for path tracing.

For the shortest path trace to succeed, the total calculated resistance cannot be greater than the value set for the maximum resistance or less than the value set for the minimum resistance. See [tpm\\_tracesetminres](#). See [tpm\\_traceshort](#).



## tpm\_tracesetminres

### [Network Tracing Functions](#)

---

Sets minimum resistance for the shortest path algorithm.

int

```
tpm_tracesetminres(  
    ade_id trace_id,  
    ads_real minres);
```

Returns **RTNORM** or an error code.

**trace\_id**      The tracing model ID, returned from [tpm\\_tracealloc](#).

**minres**        The minimum accumulated allowed path resistance.

This function sets an accumulated resistance value for path tracing. For the shortest path trace to succeed, the total calculated resistance cannot be greater than the value set for the maximum resistance or less than the value set for the minimum resistance. See [tpm\\_tracesetmaxres](#). See [tpm\\_traceshort](#).



## tpm\_traceshort

### [Network Tracing Functions](#)

---

Calculates the shortest path between two nodes.

```
int  
tpm_traceshort(  
    ade_id trace_id,  
    ade_id start,  
    ade_id end);
```

Returns **RTNORM** or an error code.

**trace\_id**     The tracing model ID, returned by [tpm\\_tracealloc](#).  
**start**         The element ID of the start node.  
**end**            The element ID of end node.

For the shortest path trace to succeed, the total calculated resistance cannot be greater than the value set for the maximum resistance or less than the value set for the minimum resistance. See [tpm\\_tracesetminres](#) and [tpm\\_tracesetmaxres](#). The accumulated resistance value is the total resistance of the nodes and links that make up the shortest path.



## tpm\_traceshortscan

### [Network Tracing Functions](#)

---

Gets the ID of a link or node in the shortest path.

```
ade_id  
tpm_traceshortscan(  
    ade_id trace_id,  
    int flag);
```

Returns a topology ID or `ADE_NULLID`.

`trace_id`     The tracing model ID, returned by [tpm\\_tracealloc](#).

`flag`         Path element code. Values can be:

- 0 Current element
- 1 First element
- 2 Last element
- 3 Next element
- 4 Previous element

First use [tpm\\_traceshort](#) to calculate a shortest path.



## tpm\_traceshortval

### [Network Tracing Functions](#)

---

Calculates the resistance of the shortest path.

```
int  
tpm_traceshortval(  
    ade_id trace_id,  
    ads_real *resist);
```

Returns **RTNORM** or an error code.

**trace\_id**      The tracing model ID returned by [tpm\\_tracealloc](#).

**resist**        The shortest path resistance.

Before calling this function, use [tpm\\_traceshort](#) to calculate the shortest path.



## Access Functions

### [Topology Function Synopsis](#)

---

The functions for accessing topologies begin with `tpm_ac`.

<a href="#">tpm_acclose</a>	Closes a topology.
<a href="#">tpm_acexist</a>	Checks if a topology exists.
<a href="#">tpm_acload</a>	Loads a topology into memory.
<a href="#">tpm_acopen</a>	Opens a topology.
<a href="#">tpm_acqty</a>	Counts topologies.
<a href="#">tpm_acunload</a>	Unloads a topology from memory.
<a href="#">tpm_acupgradeopen</a>	Changes access from read only to write.



## Analyzing Functions

### [Topology Function Synopsis](#)

---

The functions for analyzing topologies begin with `tpm_ana`.

<a href="#">tpm_anabuffer</a>	Creates a buffer space around a topology.
<a href="#">tpm_anadissolve</a>	Merges topology elements with the same value in the specified field.
<a href="#">tpm_anaoverlay</a>	Overlays two topologies.



## Building and Erasing Functions

### [Topology Function Synopsis](#)

---

The functions for topology maintenance begin with `tpm_mnt`.

<a href="#">tpm_mntbuild</a>	Builds a topology.
<a href="#">tpm_mnterase</a>	Erases a topology from the project drawing.
<a href="#">tpm_mntrebuild</a>	Rebuilds a topology.
<a href="#">tpm_mntrename</a>	Renames a topology.



## Editing Functions

### [Topology Function Synopsis](#)

---

The functions for editing topology elements begin with `tpm_edit`.

<a href="#">tpm_editaddelem</a>	Adds an element to a topology.
<a href="#">tpm_editdelelem</a>	Deletes an element from a topology.
<a href="#">tpm_editmodelem</a>	Modifies a topology element.
<a href="#">tpm_editupdelem</a>	Updates a topology element.



## Element Information Functions

### [Topology Function Synopsis](#)

---

The functions for managing topology elements begin with `tpm_elem`.

<a href="#">tpm_elemadj</a>	Compiles a list of adjacent elements.
<a href="#">tpm_elemfind</a>	Finds an element.
<a href="#">tpm_elemget</a>	Lists information about an element.
<a href="#">tpm_elemid</a>	Gets the ID of an element.
<a href="#">tpm_elemqty</a>	Counts topology elements.
<a href="#">tpm_elemss</a>	Creates a selection set of elements of a given type.



## Topology Iterating Functions

### [Topology Function Synopsis](#)

---

The functions for iterating through topologies begin with `tpm_iter`.

Many have counterparts in the topology information functions ([tpm\\_info\[xx\]](#)). The iterating functions can query any topology, loaded or unloaded, open or closed. The information functions query only topologies that are open.

<a href="#">tpm_iterdesc</a>	Gets a topology description.
<a href="#">tpm_itername</a>	Gets a topology name.
<a href="#">tpm_iternext</a>	Moves the iterator to the next topology.
<a href="#">tpm_iterstart</a>	Creates a topology iterator.
<a href="#">tpm_iterstop</a>	Frees a topology iterator.
<a href="#">tpm_itype</a>	Gets a topology type.
<a href="#">tpm_iterversion</a>	Gets the version of a topology.



## Network Tracing Functions

### [Topology Function Synopsis](#)

---

The functions for network tracing begin with `tpm_trace`.

<a href="#">tpm_tracealloc</a>	Allocates a tracing model.
<a href="#">tpm_tracebestroute</a>	Calculates the best round-trip route.
<a href="#">tpm_tracebestroutescan</a>	Gets the element ID of a link or node in the best route.
<a href="#">tpm_tracebestrouteval</a>	Calculates the resistance of the best route.
<a href="#">tpm_traceelemedit</a>	Modifies a tracing element.
<a href="#">tpm_traceelemget</a>	Gets information about a tracing element.
<a href="#">tpm_traceelemid</a>	Gets the topology ID of a tracing element.
<a href="#">tpm_traceflood</a>	Traces a flood path from a specified point.
<a href="#">tpm_tracefree</a>	Frees a tracing model.
<a href="#">tpm_traceqty</a>	Counts the selected elements after a trace.
<a href="#">tpm_tracesetmaxres</a>	Sets maximum resistance for the shortest path algorithm.
<a href="#">tpm_tracesetminres</a>	Sets minimum resistance for the shortest path algorithm.
<a href="#">tpm_traceshort</a>	Calculates the shortest path between two nodes.
<a href="#">tpm_traceshortscan</a>	Gets the topology ID of a link or node in the shortest path.
<a href="#">tpm_traceshortval</a>	Calculates the resistance of the shortest path.



## Topology Information Functions

### [Topology Function Synopsis](#)

---

The functions for getting information about topologies begin with `tpm_info`.

Many have counterparts in the topology iterating functions ([tpm\\_iter\[xx\]](#)). The information functions can query only topologies that are open. The iterating functions can query any topology, loaded or unloaded, open or closed.

<a href="#">tpm_infobuildvar</a>	Stores the configuration values of a topology.
<a href="#">tpm_infocomplete</a>	Tests if a topology is complete.
<a href="#">tpm_infocorrect</a>	Tests if a topology is correct.
<a href="#">tpm_infocurrent</a>	Checks the source from which a topology was loaded.
<a href="#">tpm_infodesc</a>	Gets a topology description.
<a href="#">tpm_infomodified</a>	Checks if topology elements have been modified using drawing tools.
<a href="#">tpm_infoname</a>	Gets a topology name.
<a href="#">tpm_infostatus</a>	Checks whether a topology is open for Read or Write.
<a href="#">tpm_infotype</a>	Gets a topology type.
<a href="#">tpm_infoversion</a>	Gets a topology version.



## Topology Query Functions

### [Topology Function Synopsis](#)

---

The functions for querying topologies begin with `tpm_qry`.

<a href="#">tpm_qrygetresdesc</a>	Gets the description of the query result topology.
<a href="#">tpm_qrygetrestopo</a>	Gets the name of the query result topology.
<a href="#">tpm_qrygettoponame</a>	Gets the name of the query source topology.
<a href="#">tpm_qrysetrestopo</a>	Defines or or undefines a query result topology.
<a href="#">tpm_qrysettoponame</a>	Defines or undefines a topology query.



## Topology Variables Functions

### [Topology Function Synopsis](#)

---

The functions for managing configuration variables begin with `tpm_var`.

<a href="#">tpm_varalloc</a>	Allocates a set of configuration variables.
<a href="#">tpm_varfree</a>	Frees a set of configuration variables.
<a href="#">tpm_varget</a>	Gets the value of a configuration variable.
<a href="#">tpm_varlist</a>	Gets all the values in a set of configuration variables.
<a href="#">tpm_varset</a>	Sets the value of a configuration variable.



## New Functions for AcMapProject Class

[Classes and Namespaces](#)

---

The [AcMapProject](#) class has new functions added for AutoCAD Map 3D 2005.

Library `rxapimap.lib`

Include `MapProj.h`

### Members

[CreatePropertyAlterationDefinition Function 1](#) [CreatePropertyAlterationDefinition Function 2](#)



## New Functions for AcMapAttachedDrawing Class

[Classes and Namespaces](#)

---

The [AcMapAttachedDrawing](#) class has a new function added for AutoCAD Map 3D 2005.

Library rxapimap.lib

Include MapArxApi.h

### Members

[ApplyThisQuery](#) Function



## New Functions for AcMapExpression Class

[Classes and Namespaces](#)

---

The [AcMapExpression](#) class has a new function added for AutoCAD Map 3D 2005.

Library `rxapimap.lib`

Include `MapArxApi.h`

### Members

[TopologyExecute Function](#)



## New Functions for AcMapQuery Class

[Classes and Namespaces](#)

---

The [AcMapQuery](#) class has new functions added for AutoCAD Map 3D 2005.

Library rxapimap.lib

Include MapArxApi.h

### Members

[Execute Function 1](#) [Execute Function 2](#)

[Execute Function 3](#)

[FileIn Function](#)

[FileOut Function](#)

[GetStringToDisplay Function](#)

[GetTopologyName Function](#)

[IsDefined Function](#)

[Select Function](#)

[SetTopologyName Function](#)



## New Functions for AcMapPropertyAlterationDefinition Class

[Classes and Namespaces](#)

---

The [AcMapPropertyAlterationDefinition](#) class has new functions added for AutoCAD Map 3D 2005.

Library `rxapimap.lib`

Include `MapAlteration.h`

### Members

[Apply Function 1](#) [Apply Function 2](#)

[Erase Function](#)

[ObjectId Function](#)



## New Functions for AcMapHatchAlteration Class

[Classes and Namespaces](#)

---

The [AcMapHatchAlteration](#) class has new functions added for AutoCAD Map 3D 2005.

Library `rxapimap.lib`

Include `MapAlteration.h`

Inherits from public `AcMapPropertyAlteration`

### Members

[Apply Function 1](#) [Apply Function 2](#)

[Erase Function](#)

[GetExcludedObjects Function](#)

[SetExcludedObjects Function](#)



## New Functions for AcMapTextAlteration Class

[Classes and Namespaces](#)

---

The [AcMapTextAlteration](#) class has new functions added for AutoCAD Map 3D 2005.

Library `rxapimap.lib`

Include `MapAlteration.h`

Inherits from public `AcMapPropertyAlteration`

### Members

[Apply Function 1](#) [Apply Function 2](#)

[Erase Function](#)



## ade\_projentitybackward

[Coordinate Transformation Functions](#)

---

Transforms an entity from the destination coordinate system to the source coordinate system.

int

```
ade_projentitybackward(  
    ads_name ent);
```

Returns RTNORM if successful.

Returns RTERROR if failed.

Returns ADEMEMERROR if memory allocation failed.

Returns ADEINVERTOR if acedInvoke() failed.

**ent**     Input entity to be transformed.

The source and destination coordinate systems must be set previously by the `ade_projsetsrc()` and `ade_projsetdest()` functions.



## ade\_projentityforward

[Coordinate Transformation Functions](#)

---

Transforms an entity from the source coordinate system to the destination coordinate system.

int

```
ade_projentityforward(  
    ads_name ent);
```

Returns RTNORM if successful.

Returns RTERROR if failed.

Returns ADEMEMERROR if memory allocation failed.

Returns ADEINVERTOR if acedInvoke() failed.

**ent**     Input entity to be transformed.

The source and destination coordinate systems must be set previously by the `ade_projsetsrc()` and `ade_projsetdest()` functions.



## ade\_projwsgeodistance

[Coordinate Transformation Functions](#)

---

Measures the geodetic distance between two points.

```
int  
ade_projwsgeodistance(  
    ads_point pt1,  
    ads_point pt2,  
    ads_real* pdDistance,  
    char** ppchDistUnits,  
    int nDistUnitsBuffSize,  
    ads_real* pdAzimuthForward,  
    ads_real* ppdAzimuthReverse);
```

Returns RTNORM if successful.

Returns RTERROR if failed.

Returns ADEMEMERROR if memory allocation failed.

Returns ADEINVERTOR if acedInvoke() failed.

<code>pt1</code>	Input first point.
<code>pt2</code>	Input second point.
<code>pdDistance</code>	Output calculated geodetic distance between the points.
<code>ppchDistUnits</code>	Output name of the distance units used in calculations. This argument is a pointer to the buffer (char**). If the buffer was not allocated, call this function with *ppchDistUnits = NULL. This function will allocate the needed buffer, and caller must free the memory that *ppchDistUnits points to. The returned display units depend on the AutoCAD Map menu setting: Map > Options > Coordinate Systems > Geodetic Distance > Units For Display.
<code>nDistUnitsBuffSize</code>	Input size of the buffer to use to copy the name of the units if *ppchDistUnits != NULL and memory was allocated when function is called.
<code>pdAzimuthForward</code>	Output angle, expressed in radians east of north, of the line from the first point to the second point, measured at the first point.
<code>ppdAzimuthReverse</code>	Output angle, expressed in radians east of north, of the line from the second point to the first point, measured at the second point.

If no coordinate system is assigned, this function uses the Pythagorean system for calculations.



## Creating Index Operation Expressions

[ade\\_dwgindex](#) [ade\\_dwgindexdef](#)

---

The following examples demonstrate creating index operation expressions.

Clears all previous index operation expressions.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("", 0, pIndexParamsRb);
```

Specifies that the location index is to be generated.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("Location", 1, pIndexParamsRb);
```

Specifies that the location index is to be removed.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("Location", 0, pIndexParamsRb);
```

Specifies that the property index is to be generated.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("Property", 1, pIndexParamsRb);
```

Specifies that the property index is to be removed.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("Property", 0, pIndexParamsRb);
```

### Object Data Indexes

If the `indexParams` argument is `RTNIL`, then indexes will be generated or removed for all tables and fields, for example:

Specifies that an object data index be generated on all tables and all fields.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("ObjData", 1, pIndexParamsRb);
```

Specifies that all object data indexes for all tables and all fields be removed.

```
struct resbuf* pIndexParamsRb = acutBuildList(RTNIL, 0);
ade_dwgindexdef("ObjData", 0, pIndexParamsRb);
```

If the **indexParams** argument includes only a table name with no field names specified, then indexes will be generated or removed for all fields in the table, for example:

Specifies that an object data index be generated on all fields in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 1, pOdIndexParamsRb);
```

Specifies that all object data indexes be removed from on all fields in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 0, pOdIndexParamsRb);
```

The list of fields in the **indexParams** argument can contain one or more fields, for example:

Specifies that an object data index be generated on field FIELD1 in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTSTR, "FIELD1",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 1, pOdIndexParamsRb);
```

Specifies that an object data index be removed from field FIELD1 in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTSTR, "FIELD1",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 0, pOdIndexParamsRb);
```

Specifies that an object data index be generated on fields FIELD1 and FIELD2 in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTSTR, "FIELD1",
    RTSTR, "FIELD2",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 1, pOdIndexParamsRb);
```

Specifies that an object data index be removed from fields FIELD1 and FIELD2 in table TABLE.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
    RTLB,
    RTSTR, "TABLE",
    RTSTR, "FIELD1",
    RTSTR, "FIELD2",
    RTLE,
    RTLE,
    0);
ade_dwgindexdef("ObjData", 0, pOdIndexParamsRb);
```

It is also valid to pass multiple table field lists through the `indexParams` parameter, for example:

Specifies creating object data indexes on all fields in TABLE1 and on FIELD1 in table TABLE2.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(
    RTLB,
```

```
        RTLB,  
        RTSTR, "TABLE1",  
        RTLE,  
        RTLB,  
        RTSTR, "TABLE2",  
        RTSTR, "FIELD1",  
        RTLE,  
        RTLE,  
    0);  
ade_dwgindexdef("ObjData", 1, pOdIndexParamsRb);
```

Specifies deleting object data indexes from all fields in TABLE1 and from FIELD1 in table TABLE2.

```
struct resbuf* pOdIndexParamsRb = acutBuildList(  
    RTLB,  
    RTLB,  
    RTSTR, "TABLE1",  
    RTLE,  
    RTLB,  
    RTSTR, "TABLE2",  
    RTSTR, "FIELD1",  
    RTLE,  
    RTLE,  
    0);  
ade_dwgindexdef("ObjData", 0, pOdIndexParamsRb);
```



## Object Data Field Types

---

The data types for object data fields are as follows. The type names in the first column are the coltype arguments for use in field definitions.

Type	Description
integer	From $-2,147,483,648$ through $2,147,483,647$
character	Any alphanumeric string, including the empty string, ""
point	List of three real numbers, separated by commas, that represent a point with an X, Y, and Z value, enclosed in quotes (a string value)
real	From $-1.7E308$ through $+1.7E308$



## Location Expressions

[ade\\_querydefine](#)

---

Location expressions are used as querycond arguments in `ade_querydefine` calls that define Location conditions.

There are a number of formats to choose from when writing Location expressions. They are listed below. The format to use depends on the Location type, which is identified in each of the following formats by the first argument. The other arguments are described below.

Location-All

`("all")`

Location-Bufferfence

`("bufferfence" searchtype offset pt1 pt2 ... ptN)`

Location-Circle

`("circle" searchtype centerpt radius)`

Location-Fence

`("fence" pt1 pt2 ... ptN)`

Location-Point

`("point" pt)`

Location-Polygon

`("polygon" searchtype pt1 pt2 ... ptN)`

Location-Polyline-Bufferfence

`("polyline" "bufferfence" searchtype offset ename)`

Location-Polyline-Fence

```
("polyline" "fence" ename)
```

Location-Polyline-Polygon

```
("polyline" "polygon" searchtype ename)
```

Location-Window

```
("window" searchtype pt1 pt2)
```

### Location Expression Parameters

searchtype	Search type keyword ( <b>string</b> ): "inside" or "crossing".
offset	Buffer offset distance ( <b>real</b> ).
ename	AutoCAD entity name, or a set of points, or "?". If "?", when the query executes, it prompts the user to click a set of points.
pt ptN centerpt	A 2D or 3D point (a list of <b>reals</b> ). If a 3D point, the Z coordinate is ignored.
radius	Radius ( <b>real</b> ).

### Location Examples

The following examples define Location conditions. The first is a Location-All condition:

```
(ade_qrydefine
  ("AND" "(" "NOT" "Location" ("All") "" )
)
```

The second is a Location-Window condition:

```
(ade_qrydefine
  ("AND" "" "" "Location"
  ("window" "crossing" (1.0 2.0) (3.0 4.0)) "")
)
```

And the third is a Location-Polyline condition.

```
(ade_qrydefine
  (list "AND" "(" "" "Location"
  (list "polyline" "bufferfence" "inside" 20.0
  (entlast)) "")
)
```

The following three examples of Location-Polyline conditions specify the `ename` argument in different ways. The first supplies an entity name:

```
(ade_qrydefine
  (list "" "" "" "Location"
    (list "polyline" "polygon" "crossing"
      (entlast) ""))
  )
```

The second supplies a set of points:

```
(ade_qrydefine
  ('"" "" "" "Location"
    ("polyline" "polygon" "crossing" (
      1 ; 1 = Polyline open, 0 = Polyline closed

      (0 0 1) ; Direction of normal vector:
        ; (0 0 1) identifies the Z axis,
        ; i.e., this polygon is parallel to
        ; the XY plane

      0.000000 ; Bulge factor of a vertex
      (4.426217 7.991379 0.000000) ; Coordinates of a vertex

      0.000000
      (2.385054 5.530788 0.000000)

      0.000000
      (4.648083 3.912562 0.000000)

      0.000000
      (5.912716 5.708128 0.000000)

      0.000000
      (7.754200 3.823892 0.000000)

      0.000000
      (8.020439 3.646552 0.000000) ) ) ""))
  )
```

And the third prompts the user to click a set of points:

```
(ade_qrydefine  
  ('"" "" "" "Location"  
   ("polyline" "polygon" "crossing" "?") ""))  
)
```



## Property Expressions

[ade\\_querydefine](#)

Property expressions are used as `querycond` arguments in `ade_querydefine` calls that define Property conditions.

They have the following format:

(property operator value [subclasses])

### Property Expression Parameters

property	Property name ( <b>string</b> ). See the Property and Value Arguments table below.
operator	"=", ">", "<", "<=", ">=", "<>". Note that the only valid operator in a string context is "=".
value	Depends on the <b>property</b> argument. See the Property and Value Arguments table below.
subclasses	Optional. <b>T</b> or <b>nil</b> . The default if the argument is omitted is <b>nil</b> . This setting has no effect unless <b>property</b> is "feature". <b>T</b> means return all objects belonging to the feature class identified by the <b>value</b> argument, including objects belonging to any subclass of that feature. <b>nil</b> means do not include objects belonging to such a subclass.

### Property and Value Arguments

Property	Value
area	Area value ( <b>string</b> ).
blockname	Block name ( <b>string</b> ).
color	<a href="#">Color</a> ( <b>string</b> ).
elevation	Z coordinate ( <b>string</b> ).
"feature"	Feature name ( <b>string</b> ).
group	Group name ( <b>string</b> ).
layer	Layer name ( <b>string</b> ).

<code>length</code>	Length ( <code>string</code> ).
<code>linetype</code>	Line type ( <code>string</code> ).
<code>"lineweight"</code>	Line weight ( <code>string</code> ).
<code>"plotstyle"</code>	Plot style ( <code>string</code> ).
<code>style</code>	Text style ( <code>string</code> ).
<code>thickness</code>	Thickness ( <code>string</code> ).
<code>objtype</code>	Object type ( <code>string</code> ), or <code>"unknown"</code> .
<code>value</code>	Text value ( <code>string</code> ).

### Property Expression Examples

The following examples define Property conditions. The first specifies a layer:

```
(setq qry_id
  (ade_qrydefine
    '("and" "" "" "property" ("layer" "=" "WATER") "" )
  )
)
```

And the second specifies a color:

```
(setq qry_id
  (ade_qrydefine
    '("or" "(" "not" "property" ("color" "=" "RED") "" )
  )
)
```



## Data Expressions

[ade\\_querydefine](#)

Data expressions are used as querycond arguments in `ade_querydefine` calls that define Data conditions.

They have the following format.

(datatype tablename.fieldname operator value [subclasses])

### Data Expression Parameters

datatype	Data type to match (string): "objdata", "attrib", "aselink", "EED", or "feature".
tablename	Depends on the datatype argument. See the Tablename And Fieldname Arguments table below.
fieldname	Depends on the datatype argument. See the Tablename And Fieldname Arguments table below.
operator	Comparison operator (string): "=", ">", "<", "<=", ">=", or "<>". Note that the only valid operator in a string context is "=".
value	Value to match.
subclasses	Optional. T or nil. The default if the argument is omitted is T. This setting has no effect unless datatype is "feature". T means return objects belonging to the feature class identified by the tablename argument, including objects belonging to any subclass of that feature. nil means do not include objects belonging to such a subclass.

The `tablename` and `fieldname` arguments depend on the `datatype` argument:

### Tablename and Fieldname Arguments

datatype	tablename	fieldname
objdata	Table name.	Field name.
attrib	Block name.	Attribute definition.
aselink	Link template.	Column name.
EED	RegApp name.	EED field name.

feature	Feature Class name.	Property name.
---------	---------------------	----------------

## Data Expression Examples

The following examples define Data conditions of various types.

```
(ade_qrydefine
  '(" "" "" "" "Data"
    ("attrib" "*.Type" "=" "c*")"))

(ade_qrydefine
  '(" "" "" "" "Data"
    ("objdata" "mytable.fl1" "=" "1")"))

(ade_qrydefine
  '(" "" "" "" "Data"
    ("asmlink" "cpu_lpn.cpu" "=" "MAC2LC")"))

(ade_qrydefine
  '(" "" "" "" "Data"
    ("EED" "REGAPP.STREET" "=" "Willow")"))

(ade_qrydefine
  '(" "" "" "" "Data"
    ("EED" "REGAPP.#NUMBER" "=" "512")"))

(ade_qrydefine
  '(" "" "" "" "Data"
    ("EED" "REGAPP.&SQLLINK;" "=" "MAC2LC")"))
```

Note in the last example that **&SQLLINK** is not treated as a link template key. It is treated just like any other ADE 1.0 EED field. For example, if an object has EED such as

```
(-3 (1000 . "&EEDFIELD;" = "1234"))
```

then the Data condition to retrieve the object is written as follows:

```
("EED" "REGAPP.&EEDFIELD;" = "1234")
```

If the EED is defined by

```
(-3 (1000 . "&EEDFIELD;" = "'First', 'Last'"))
```

then the Data condition to retrieve the object is written as follows:

```
("EED" "REGAPP.&EEDFIELD;" "=" "'First', 'Last'")
```

In other words, everything after the equal sign is treated as one string. That way you can use any pattern (for wcmatch) in the query.



## SQL Expressions

[ade\\_qrydefine](#)

---

SQL expressions are used as `querycond` arguments in `ade_querydefine` calls that define SQL conditions.

They have the following format:

```
(linkpathname sqlcondition)
```

### SQL Expression Parameters

`linkpathname`      Link template (**string**).

`sqlcondition`      SQL condition (**string**).

The `sqlcondition` argument should contain only the **WHERE** clause of an SQL statement (for example, `"last_name = 'Smith'"`). To select the entire table, let the `sqlcondition` argument be the empty string (`""`).

```
(ade_qrydefine  
  ("AND" "" "" "Sql"  
   ("EMPLN3" "" "" )  
  )
```



## Range Table Operators

[ade\\_rtdfrange](#)

---

You can use the following comparison operators in range table expressions:

< > <= >= /= otherwise

When the first element in the range expression is "otherwise", the second element must be the empty string (""). For example: ("otherwise" "" "red"). Although you do not specify a comparison value when you use the **otherwise** operator, a second list element is still required.

**Note** The not-equal operator, represented here by "=", is represented by "<>" in other Data Extension functions.

Links

[DMScaleFactorReactor Class](#), [DMScaleFactorReactor Class](#),  
[AcMapDMScaleFactor Namespace](#)

DMScaleFactorReactor::~~DMScaleFactorReactor Destructor  
[DMScaleFactorReactor Class](#) | [DMScaleFactorReactor Class](#) |  
[AcMapDMScaleFactor Namespace](#)

Destroys an instance of this class.

`~DMScaleFactorReactor();`

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[DMScaleFactorReactor Class](#), [DMScaleFactorReactor Class](#),  
[AcMapDMScaleFactor Namespace](#)

DMScaleFactorReactor:: DMScaleFactorReactor Constructor  
[DMScaleFactorReactor Class](#) | [DMScaleFactorReactor Class](#) |  
[AcMapDMScaleFactor Namespace](#)

Constructs an instance of this class.

```
DMScaleFactorReactor();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[DMScaleFactorReactor Class](#), [DMScaleFactorReactor Class](#),  
[AcMapDMScaleFactor Namespace](#)

DMScaleFactorReactor:: ScaleFactorChanged Method  
[DMScaleFactorReactor Class](#) | [DMScaleFactorReactor Class](#) |  
[AcMapDMScaleFactor Namespace](#)

Invoked when the scale factor changes.

```
virtual void ScaleFactorChanged() = 0;
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: ~AcMapContainer Destructor

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Destroys an instance of this class.

**virtual** ~AcMapContainer();

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: AcMapContainer Constructor

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Constructs an instance of this class.

```
AcMapContainer();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: GetContainerIterator Method

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Returns the container iterator for this container.

```
virtual Acad::ErrorStatus GetContainerIterator(  
    AcMapContainerIterator*& pContIter  
) const = 0;
```

Parameters

Description

pContIter

Output container iterator pointer. The container iterator should be deleted when no longer needed.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code in which case the returned iterator will be NULL..

Remarks

Since the container represents a tree node, the container iterator it returns are the child and siblings of the node which do not directly contain tiles, but rather have children.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: GetTileIterator Method

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Returns the tile iterator for this container.

```
virtual Acad::ErrorStatus GetTileIterator(  
    AcMapTileIterator*& pTileIter  
) const = 0;
```

Parameters

Description

pTileIter

Output tile iterator pointer. The tile iterator should be deleted when no longer needed.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code in which case the returned iterator will be NULL.

Remarks

Since the container represents a tree node, the tile iterator it returns are the child and siblings of the node which do not directly contain tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: Name Method

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Sets the Node name for the node associated with this container.

```
virtual const ACHAR* Name() const = 0;
```

Returns

Returns the node name, or the empty string if the node has no name.

## Remarks

Typically a node will have a name if it does not have a tile. For instance in a grid tiling scheme, typically the first level below the root will contain the rows. Each row will contain a child whose siblings contain tiles for each column in the row. The row nodes will typically be named with the Row name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: SetName Method

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Sets a node name for the node associated with this container.

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR * pszName  
) = 0;
```

Parameters

Description

pszName

Input tile name. The name should specify a valid node name. Node names follow esn rules.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Node names are typically used where the node represents the row or column or other subset of the tile name contained somewhere below the level of this container's node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainer Class](#), [AcMapMbTileSet::AcMapContainer Class](#)

AcMapMbTileSet:: AcMapContainer:: TreeNode Method

[AcMapMbTileSet::AcMapContainer Class](#) |

[AcMapMbTileSet::AcMapContainer Class](#)

Returns the node associated with this container.

```
virtual AcMapTreeNode* TreeNode() const = 0;
```

Returns

Returns the tree node for this container. If it fails for any reason it returns NULL.

Remarks

Any node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator:: ~AcMapContainerIterator  
Destructor

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Destroys an instance of this class.

**virtual** [~AcMapContainerIterator\(\)](#);

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator::

AcMapContainerIterator Constructor

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Constructs an instance of this class.

```
AcMapContainerIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator:: Done Method

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Tells the user if we are done.

```
virtual bool Done() const = 0;
```

Returns

Returns true if done; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator:: Get Method

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Returns the next node in the chain as a container.

```
virtual bool Get(  
    const AcMapContainer*& pContainer  
) const = 0;
```

Parameters	Description
pContainer	Output container.

Returns

Returns true if successful; otherwise, returns false to indicate there are no more containers.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator:: Length Method

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Returns the number of containers in the iterator.

```
virtual unsigned int Length() const = 0;
```

Returns

Returns the size of the iterator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),  
[AcMapMbTileSet::AcMapContainerIterator Class](#)  
AcMapMbTileSet:: AcMapContainerIterator:: Rewind Method  
[AcMapMbTileSet::AcMapContainerIterator Class](#) |  
[AcMapMbTileSet::AcMapContainerIterator Class](#)

Resets the iterator.

```
virtual bool Rewind(  
    bool bReversed = false  
) const = 0;
```

Parameters	Description
bReversed	Direction of iteration.

Returns

Returns true if successful; otherwise, returns false.

Remarks

Sets the iteration to go either forward or reversed. Default is forward.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapContainerIterator Class](#),

[AcMapMbTileSet::AcMapContainerIterator Class](#)

AcMapMbTileSet:: AcMapContainerIterator:: Step Method

[AcMapMbTileSet::AcMapContainerIterator Class](#) |

[AcMapMbTileSet::AcMapContainerIterator Class](#)

Moves to the next node in the chain as a container.

```
virtual bool Step() const = 0;
```

Returns

Returns true if successful; otherwise, returns false to indicate there are no more containers.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: ~AcMapTileIterator Destructor

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTileIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: AcMapTileIterator Constructor

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Constructs an instance of this class.

```
AcMapTileIterator();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: Done Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Tells the user if we are done.

```
virtual bool Done() const = 0;
```

Returns

Returns true if done; otherwise, returns false.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: Get Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Returns the current tile in the chain.

```
virtual bool Get(  
    const AcMapMbTile*& pTile  
) const = 0;
```

Parameters	Description
------------	-------------

pTile	Output Tile.
-------	--------------

Returns

Returns true if successful; otherwise, returns false to indicate there are no more tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: GetNode Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Returns the node associated with the current tile in the chain.

```
virtual bool GetNode(  
    AcMapMbTileSet::AcMapTreeNode*& pNode  
) const = 0;
```

Parameters	Description
------------	-------------

pNode	Output Tile.
-------	--------------

Returns

Returns true if successful; otherwise, returns false to indicate there are no more nodes.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: Length Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Returns the number of containers in the iterator.

```
virtual unsigned int Length() const = 0;
```

Returns

Returns the size of the iterator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: Rewind Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Resets the iterator.

```
virtual bool Rewind(  
    bool bReversed = false  
) const = 0;
```

Parameters	Description
bReversed	Direction of iteration.

Returns

Returns true if successful; otherwise, returns false.

Remarks

Sets the iteration to go either forward or reversed. Default is forward.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator::

SetCurrentSecondaryIndexName Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Sets a node name for the node associated with this tile.

```
virtual Acad::ErrorStatus SetCurrentSecondaryIndexName(  
    const ACHAR * pszName  
) = 0;
```

Parameters

Description

pszName

Input tile name. The name should specify a valid node name. Node names follow esn rules.

Returns

Returns true if successful; otherwise, returns false.

Remarks

This function is meant to handle grid tiling schemes where there is no place to store the value of the secondary index. The primary index is the container's name for the current tile iterator.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTileIterator Class](#),

[AcMapMbTileSet::AcMapTileIterator Class](#)

AcMapMbTileSet:: AcMapTileIterator:: Step Method

[AcMapMbTileSet::AcMapTileIterator Class](#) |

[AcMapMbTileSet::AcMapTileIterator Class](#)

Moves to the next tile in the chain.

```
virtual bool Step() const = 0;
```

Returns

Returns true if successful; otherwise, returns false to indicate there are no more tiles.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: ~AcMapTreeNode Destructor

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Destroys an instance of this class.

```
virtual ~AcMapTreeNode();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: AcMapTreeNode Constructor

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Constructs an instance of this class.

```
AcMapTreeNode();
```

Returns

Returns nothing.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: AddChild Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets a child for this node.

```
virtual Acad::ErrorStatus AddChild(  
    AcMapTreeNode * pChild,  
    bool bAtEnd = false  
) = 0;
```

Parameters

Description

pChild

Input tree node.

bAtEnd

Input indicating where to put the added child. If true, then put the child at the end of the chain of the child of this node. If false, then insert the node as the child of this node and make the current child the first sibling in the chain.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

If there is already an existing non null child for this node, then this function will place the node at the beginning or end of the chain of siblings of the child of this node.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet::AcMapTreeNode::AddSibling Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets a sibling for this node.

```
virtual Acad::ErrorStatus AddSibling(  
    AcMapTreeNode * pSibling,  
    bool bAtEnd = false  
) = 0;
```

Parameters

Description

pSibling

Input sibling.

bAtEnd

Input indicating where to put the added sibling. If true, then put the sibling at the end of the chain of siblings, if false, then insert the node as the first sibling in the chain.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

If the node already has a sibling, then this node is inserted into the sibling chain.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: Child Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the child of this node.

```
virtual AcMapTreeNode* Child() const = 0;
```

Returns

Returns the tree node if there is a child to this node. otherwise, returns NULL.

## Remarks

Any node contains one or zero children. The child may have siblings however.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: ConstChild Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the const child of this node.

```
virtual const AcMapTreeNode* ConstChild() const = 0;
```

Returns

Returns the tree node if there is a child to this node. otherwise, returns NULL.

## Remarks

Any node contains one or zero children. The child may have siblings however.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: ConstSibling Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the const sibling of this node.

```
virtual const AcMapTreeNode* ConstSibling() const = 0;
```

Returns

Returns the tree node if there is a sibling to this node. otherwise, returns NULL.

## Remarks

Once the tile set has been constructed, the tile set should be accessed thru const member functions.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: ConstTile Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the const tile of this node.

```
virtual const AcMapMbTile* ConstTile() const = 0;
```

Returns

Returns the tile if this node has a tile. otherwise, returns NULL.

## Remarks

A tree node must contain either a child or a tile. But never contains both.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: Name Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets the Node name.

```
virtual const ACHAR* Name() const = 0;
```

Returns

Returns the node name, or the empty string if the node has no name.

## Remarks

Typically a node will have a name indicating the grouping for the tiles below it. For instance in a grid tiling scheme, typically the first level below the root will contain the rows. Each row will contain a child whose siblings contain tiles for each column in the row. The row nodes will typically be named with the Row name.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: SetName Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets a node name.

```
virtual Acad::ErrorStatus SetName(  
    const ACHAR * pszName  
) = 0;
```

Parameters

Description

pszName

Input tile name. The name should specify a valid node name. Node names follow esn rules.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Remarks

Node names may be used where the name indicates the category of the tiles below it in the tree.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: SetSibling Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets a sibling for this node.

```
virtual Acad::ErrorStatus SetSibling(  
    AcMapTreeNode * pSibling  
) = 0;
```

Parameters	Description
pSibling	Input sibling.

Returns

Returns true if successful; otherwise, returns false.

## Remarks

If the node already has a sibling, then this node is inserted into the sibling chain.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: SetTile Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Sets the tile for this node.

```
virtual Acad::ErrorStatus SetTile(  
    AcMapMbTile * pTile  
) = 0;
```

Parameters

Description

pTile

Input tile. Current tile if any is discarded.

Returns

Returns Acad::eOk if successful; otherwise, returns a different error code.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: Sibling Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the sibling of this node.

```
virtual AcMapTreeNode* Sibling() const = 0;
```

Returns

Returns the tree node if there is a sibling to this node. otherwise, returns NULL.

## Remarks

Once the tile set has been constructed, the tile set should be accessed thru const member functions. These functions might be used during construction.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

## Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: Tile Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the tile of this node.

```
virtual AcMapMbTile* Tile() const = 0;
```

Returns

Returns the tile if this node has a tile. otherwise, returns NULL.

## Remarks

A tree node must contain either a child or a tile. But never contains both.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).

Links

[AcMapMbTileSet::AcMapTreeNode Class](#), [AcMapMbTileSet::AcMapTreeNode Class](#)

AcMapMbTileSet:: AcMapTreeNode:: TileSet Method

[AcMapMbTileSet::AcMapTreeNode Class](#) |

[AcMapMbTileSet::AcMapTreeNode Class](#)

Returns the owner tile set of this node.

```
virtual const AcMapMbTileSet* TileSet() const = 0;
```

Returns

Returns the tile set if successful otherwise, returns NULL.

Created with a commercial version of [Doc-O-Matic](#). In order to make this message disappear you need to register this software. If you have problems registering this software please contact us at [support@toolsfactory.com](mailto:support@toolsfactory.com).



## Plot Set Attributes

This table shows plot set attribute names, descriptions, examples, and default values.

Attribute	Description	Default
name	Query name ( <i>string</i> ), e.g., "plotset2"	""
desc	Description ( <i>string</i> ), e.g., "Complete Plot Set"	""
pnam	Plotter name ( <i>string</i> ), e.g., "HP LaserJet III on LPT1"	""
pscr	Plot script ( <i>string</i> ), e.g., "e, n, y"	""
block	Name of plot layout, a block name ( <i>string</i> ), e.g., "TBLOCK"	""
atts	List of attributes for the block ( <i>string list</i> ), e.g., "TITLE"	nil
vlayer	Main viewport layer ( <i>string</i> ), e.g., "VPORT2"	""
vdysl	Layers to disable for the main viewport ( <i>string list</i> ), e.g., "0"	nil
kflg	Enables the reference viewport: <b>T</b> or <b>nil</b> <b>T</b> Enabled <b>nil</b> Disabled Note that there are two dependent attributes, "klayer" and "kdispl"	<b>T</b>
klayer	Key viewport layer ( <i>string</i> ), e.g., "VPORT1", the layer of the reference viewport, which must be a different layer from the layer used for the main viewport (attribute "vlayer"); if the function <code>map_pltblkvps</code> returns only one viewport layer, it should be used for the main viewport	""
kdispl	Key display layers ( <i>string list</i> ), e.g., "BOUNDARY", a list of layers to be displayed in the referenced viewport. This should not be <b>nil</b> if "kflg" is enabled. Valid layers are all the layers returned by <code>ade_dsproplist</code> using the option "layer"	nil
kscl	Key viewport scale factor ( <i>real</i> )	0.25
dwgs	List of source drawings ( <i>string list</i> ), e.g., "DRAWING1.DWG" "DRAWING2.DWG"	nil

dact	Force drawing active at plot time: <b>T</b> or <b>nil</b>	<b>T</b>
qcat	List of query catalogs ( <b>string</b> list), paired with <b>qnam</b> list entries, e.g., "Queries"	<b>nil</b>
qnam	List of query names ( <b>string</b> list), paired with <b>qcat</b> list entries, e.g., "Query_One"	<b>nil</b>
bdwg	Boundary drawing ( <b>string</b> ), e.g., "BOUNDARY.DWG"	""
bnds	Boundaries ( <b>string</b> list), the boundary or boundaries to use by providing the value for the field assigned in "bnamf", e.g., "Boundary #3"	<b>nil</b>
blyr	Boundary layer ( <b>string</b> ), the name of the layer the boundary is placed on. It can be any layer of the boundary drawing. For example, "BOUNDARY"	""
bnamt	Boundary name table ( <b>string</b> ), the name of the object data table attached to the boundary object, e.g., "BOUNDARIES"	""
bnamf	Boundary name field ( <b>string</b> ), the name of the field to use (from the table assigned in "bnamt") for the "bnds" attribute, e.g., "name"	""
bodfs	Boundary fields ( <b>string</b> list), the list of field names to be mapped to attributes in the title block, e.g., "name" (see details and example following this table); the title block is assigned using the attributes from the block assigned in "block"	<b>nil</b>
sflg	Plot to specified scale: <b>T</b> or <b>nil</b>	<b>nil</b>
scale	Plot scale ( <b>string</b> ), e.g., "1:2000"	1:1
clip	Clip objects against boundary (flag): <b>T</b> or <b>nil</b>	<b>nil</b>
pbnd	Plot boundary (flag): <b>T</b> or <b>nil</b> <b>T</b> Plot boundary polylines are plotted <b>nil</b> Plot boundary polylines are erased before plotting	<b>nil</b>
bbuf	Buffer boundary: <b>T</b> or <b>nil</b>	<b>nil</b>
btyp	Buffer type: "true" or "rect" "true" Offsets the minimum bounding rectangle for a map boundary polyline "rect" Specifies an offset of an existing plot boundary line	<b>true</b>
bdist	Buffer distance ( <b>real</b> )	0.10

Mapping of object data fields to block attributes is managed by the two plot attributes **"bodfs"** and **"atts"**. The mapping is best explained by example. Suppose the following conditions:

- The boundary object data table, defined by the **"bnamf"** attribute, has the following fields: Field1, Field2, Field3, and Field4.
- The layout block, defined by the **"block"** attribute, has the following attributes: Attr1, Attr2, Attr3, Attr4, and Attr5.
- You want the value of Field1 to appear in place of Attr2, the value of Field2 in place of Attr3, and value of Field4 in place of Attr1.

To define the desired mapping, set the **"bodfs"** attribute to (**"field1" "field2" "field4"**) and the **"atts"** attribute to (**"attr2" "attr3" "attr1"**). When plotting is done, values of the fields from the object data table attached to the boundary's closed polyline will be assigned to the specified block attributes. The number of elements in each list for **"bodfs"** and **"atts"** must be the same.

## AcMapOSEObject::AddToEditSet

[AcMapOSEObject class](#)

Adds the object's AutoCAD counterpart to the EditSet.

```
bool  
AddToEditSet();
```

Returns True if successful.

If an object is added to this user's EditSet, it is locked for other users.

## AcMapOSEObject::FeatureName

[AcMapOSEObject class](#)

Gets the object's feature type.

```
const char *  
FeatureName();
```

Returns the feature type.

# AcMapOSEObject::IsErased

[AcMapOSEObject class](#)

Tells if the object's AutoCAD counterpart has been erased.

bool

IsErased();

Returns True if the AutoCAD counterpart has been erased.

# AcMapOSEObject::IsInEditSet

[AcMapOSEObject class](#)

Tells if the object's AutoCAD counterpart is in the EditSet.

```
bool  
IsInEditSet();
```

Returns true if the AutoCAD counterpart is in the EditSet.

If the object is in this user's EditSet, it is locked for other users.

# AcMapOSEObject::IsModified

[AcMapOSEObject class](#)

Tells if the object's AutoCAD counterpart has been modified.

```
bool  
IsModified();
```

Returns true if the AutoCAD counterpart has been modified.

# AcMapOSEObject::IsUpToDate

[AcMapOSEObject class](#)

Tells if the object is up to date.

```
bool  
IsUpToDate();
```

Returns true if the object is up to date.

An object is up to date if it's AutoCAD and Oracle counterparts have the same version number.

# AcMapOSEObject::RemoveFromEditSet

[AcMapOSEObject class](#)

Removes the object's AutoCAD counterpart from the EditSet.

```
bool  
RemoveFromEditSet();
```

Returns true if successful.

If an object is removed from this user's EditSet, it is unlocked for other users.

# AcMapOSEObject::Version

[AcMapOSEObject class](#)

Gets the version number of the object's AutoCAD counterpart.

unsigned long  
Version();

Returns the version number, an integer starting with 1.

# AcMapOSEObject::WhoHasIt

[AcMapOSEObject class](#)

Gets information about the user who has this object in their EditSet.

```
bool  
WhoHasIt(  
    unsigned long & UserId,  
    std::string & strNtUser,  
    std::string & strDbUser,  
    std::string & strMapUser,  
    std::string & strComputer,  
    std::string & strLoginTime);
```

Returns True if successful.

<code>UserId</code>	Owner's user ID.
<code>strNtUser</code>	Owner's NT user name.
<code>strDbUser</code>	Owner's Oracle user name.
<code>strMapUser</code>	Owner's AutoCAD Map user name.
<code>strComputer</code>	Owner's computer name.
<code>strLoginTime</code>	Owner's log in time.

If this user cannot add an object to their EditSet, it may be in someone else's EditSet. WhoHasIt() gets information about the object's owner.

# AcMapOSEProject::AddToEditSet

[AcMapOSEProject class](#)

Adds drawing objects to the EditSet.

```
bool  
AddToEditSet(  
    const AcDbObjectIdArray & acadIds,  
    AcDbObjectIdArray & arrFilteredOut);
```

Returns true if successful.

**acadIds**            Selection set of objects to add.

**arrFilteredOut**    Selection set of objects that could not be added. Normally because the objects were in the EditSet already, or they were in some other user's EditSet, or they were not queried objects.

# AcMapOSEProject::FilterNewObjects

[AcMapOSEProject class](#)

Given a selection set of drawing objects, compiles lists of which are new (and of a certain feature type, or types), and which are not.

```
bool  
FilterNewObjects(  
    const std::vector<_string class="calibre8"> & vFeatures,  
    const AcDbObjectIdArray & arrInput,  
    AcDbObjectIdArray & arrNew,  
    AcDbObjectIdArray & arrFilteredOut);
```

Returns true if successful.

<a href="#">vFeatures</a>	Input list of feature types. The output list, arrNew, will include objects of these feature types only. If the vFeatures list is empty, the output list will include objects of any valid feature type.
<a href="#">arrInput</a>	Input selection set of drawing objects to filter. If the arrInput list is empty, the selection set includes all objects in the drawing.
<a href="#">arrNew</a>	Output list of arrInput objects that are new and of vFeatures feature type(s).
<a href="#">arrFilteredOut</a>	Output list of arrInput objects not included in arrNew.

Objects in the project drawing are considered new if they are valid features and they are not in the project's Oracle database. An object is a valid feature if it resides on a drawing layer associated with a feature type.

# AcMapOSEProject::FilterQueriedObjects

[AcMapOSEProject class](#) [See Also](#)

Given a selection set of drawing objects, compiles lists of which are queried (and of a certain feature type, or types, and EditSet status), and which are not.

```
bool  
FilterQueriedObjects(  
    const std::vector<_string class="calibre8"> & vFeatures,  
    const AcDbObjectIdArray & arrInput,  
    AcDbObjectIdArray & arrErased,  
    AcDbObjectIdArray & arrModified,  
    AcDbObjectIdArray & arrUnchanged,  
    AcDbObjectIdArray & arrFilteredOut,  
    int nOptions);
```

Returns true if successful.

<b>vFeatures</b>	Input list of feature types. The output lists, arrErased, arrModified, and arrUnchanged, will include objects of these feature types only. If the vFeatures list is empty, the output list will include objects without regard to feature type.
<b>arrInput</b>	Input selection set of drawing objects to filter. If the arrInput list is empty, the selection set includes all objects in the drawing.
<b>arrErased</b>	Output list of arrInput objects that are queried and erased, and of vFeatures feature type(s) and nOptions edit-set status.
<b>arrModified</b>	Output list of arrInput objects that are queried and modified, and of vFeatures feature type(s) and nOptions edit-set status.
<b>arrUnchanged</b>	Output list of arrInput objects that are queried and unchanged, and of vFeatures feature type(s) and nOptions edit-set status.
<b>arrFilteredOut</b>	Output list of arrInput objects not included in arrErased, arrModified, or arrUnchanged.
<b>nOptions</b>	Input edit-set status options, a bitwise OR combination of AcMapOSEProject::EProjectOptions enumerators expressing whether output lists should include only objects in the EditSet, only objects not in the EditSet, or both.

Objects in the project drawing are considered queried if they are in the project's Oracle database. Objects in this user's EditSet are locked for other users. Objects not in this user's EditSet are unlocked for other users.

For information about edit-set status options ("project options"), see [AcMapOSEProject::EProjectOptions enumeration](#).



# AcMapOSEProject::RemoveFromEditSet

[AcMapOSEProject class](#)

Removes drawing objects from the EditSet.

```
bool  
RemoveFromEditSet(  
    const AcDbObjectIdArray & acadIds,  
    AcDbObjectIdArray & arrFilteredOut);
```

Returns true if successful.

**acadIds**            Input selection set of drawing objects to remove.

**arrFilteredOut**    Output selection set of objects that could not be removed. Normally because they were not in the EditSet to begin with.

# AcMapOSEProject::EProjectOptions enumeration

[AcMapOSEProject class](#)

Edit-set status options for queried objects.

0x1	kInEditSet	Object is in this user's EditSet (i.e., it is locked). Only this user can edit it.
0x2	kNotInEditSet	Object is not in this user's EditSet (i.e., it is unlocked). Any user can lock it for editing.

# Importing from an Oracle Spatial Database

The following sample illustrates importing data from an Oracle Spatial database to a project drawing. It is in two parts. First, it defines a query to specify a condition for the objects to import, and then it imports them.

**Note** The sample assumes that there is an open project drawing, and that an import query has been defined already through the Import dialog box in the UI (choose Map > OracleSpatial > ImportObjectsFromOracle). It also assumes that the connection object is actually connected to a database. Click "Connecting to an Oracle Spatial Database" for sample code.

[Connecting to an Oracle Spatial Database \(sample\)](#) [Querying and importing data \(overview\)](#)  
[Related samples](#)

```
#include "StdAfx.h"
#include "StdArx.h"
#include "AcMapOracleConnection.h"
#include "AcMapOracleQuery.h"
#include "AcMapOracleImport.h"

BOOL ImportFromOracle()
{
    // Get the connection object
    AcMapOracleConnection *pConnection = AcMapOracleGetConnection();

    /***** Query *****/

    // Instantiate a query object
    AcMapOracleQuery *pQuery =
        new AcMapOracleQuery(pConnection);

    // Initialize query object with current import query,
    // which has been defined already through the UI
    pQuery->InitWithCurrent(
        acdbHostApplicationServices()->workingDatabase());

    // Or initialize query with specific condition
    pQuery->Init("EntityType = 'AcDbCircle'");

    // You can check the whole query by
    // using ConvertToSql function
    char *pSql = pQuery->ConvertToSql();
    acutPrintf("\nQuery in SQL format:%s", pSql);

    // Save the query to the Oracle Spatial database
    pQuery->Save("MyQuery");

    // Load the saved query in a new query object
    AcMapOracleQuery *pNewQuery =
        new AcMapOracleQuery(pConnection);
}
```

```
pNewQuery->Load("MyQuery");

// Compare old and new query
char *pNewSql = pNewQuery->ConvertToSql();

if(strcmp(pSql, pNewSql) == 0)
    acutPrintf("Query Load/Save works!!\n");
else
    acutPrintf("Query Load/Save failed!!\n");

/*****
/***** Import *****/

// Instantiate an import object
AcMapOracleImport *pImport =
    new AcMapOracleImport(pConnection);

// Use the query object defined above to
// import Oracle Spatial data
if(pImport->CheckImport())
{
    if(pImport->Import(pQuery, false))
        acutPrintf("\nImported successfully\n");
    else
        acutPrintf("\nImport failed\n");
}
else
    acutPrintf("\nCheckImport() failed, can not import!\n");
}
```

# Connecting to an Oracle Spatial Database

The following code sample illustrates connecting to an Oracle Spatial database and adding custom reactors. Custom reactor declarations are in MyMapOracleReactors.h, one of the files included at the beginning of the sample. The "Subclassing Custom Reactors" sample shows what such a file would contain.

[Managing the connection \(overview\)](#) [Subclassing Custom Reactors \(sample\)](#)  
[Other related samples](#)

```
#include "StdAfx.h"
#include "StdArx.h"
#include "AcMapOracleConnection.h"
#include "MyMapOracleReactors.h"      // Custom reactors

BOOL ConnectToOracle()
{
    // Get the connection object
    AcMapOracleConnection *pConnection = AcMapOracleGetConnection();

    // Initialize reactors
    // Custom reactor types are declared in MyMapOracleReactors.h
    MyConnectionReactor *pConnectReactor = new MyConnectionReactor();
    MyExportReactor *pExportReactor = new MyExportReactor();
    MyImportReactor *pImportReactor = new MyImportReactor();

    // Add reactors to connection class
    pConnection->AddConnectionReactor(pConnectReactor);
    pConnection->AddExportReactor(pExportReactor);
    pConnection->AddImportReactor(pImportReactor);

    BOOL bConnect = FALSE;

    // If no database is connected, connect one
    if(!pConnection->IsConnected())
    {
        bConnect = pConnection->Connect("Scott","TestDB","Scott","Tiger");
    }
    else
    {
        acutPrintf("Already connected to Oracle!!");
        return FALSE;
    }

    // Check if schema is valid
    if(bConnect)
    {
        if(pConnection->IsSchemaValid())
            acutPrintf("\nSchema is valid for this connection\n");
        else
            acutPrintf("\nSchema is not valid\n");
    }
}
```

```
// Get the service name
char *pService = 0;
pService = new char[80];

pService = pConnection->Service();
acutPritnf("\nService is %s\n", pService);

delete [] pService;

// Get the schema name
char *pSchema = 0;
pSchema = new char[80];

pSchema = pConnection->Schema();
acutPrintf("\nSchema is %s\n", pSchema);

delete [] pSchema;

// Get the user name
char *pUserName = 0;
pUserName = new char[80];

pUserName = pConnection->UserName();
acutPrintf("\nUserName is %s\n", pUserName);

delete [] pUserName;

// Disconnect the database
if(pConnection->Disconnect())
    acutPrintf("\nDisconnected successfully from Oracle\n");
else
    acutPrintf("\nError disconnecting from Oracle\n");

//Remove reactors from connection interface
pConnection->RemoveConnectionReactor(pConnectReactor);
pConnection->RemoveExportReactor(pExportReactor);
pConnection->RemoveImportReactor(pImportReactor);

}
```

# Exporting to an Oracle Spatial Database

The following code sample illustrates exporting data from a project drawing to an Oracle Spatial database.

**Note** The sample assumes an open project drawing containing some objects. It also assumes that the connection object is actually connected to a database. Click "Connecting to an Oracle Spatial Database" for sample code.

[Connecting to an Oracle Spatial Database \(sample\)](#) [Exporting data \(overview\)](#)  
[Related samples](#)

```
#include "StdAfx.h"
#include "StdArx.h"
#include "AcMapOracleConnection.h"
#include "AcMapOracleExport.h"

BOOL ExportToOracle()
{
    // Get the connection object
    AcMapOracleConnection *pConnection = AcMapOracleGetConnection();

    //Instantiate an export object
    AcMapOracleExport *pExport = new AcMapOracleExport(pConnection);

    bool bExport = false;

    // Export 1
    // Default options...
    //   From all layers, export
    //     object data, block attributes, link templates
    //   Erase drawing objects after export
    bExport = pExport->ExportObjectsAll();

    if(bExport)
        acutPrintf("\nExport 1 successful!!\n");
    else
        acutPrintf("\nExport 1 failed!\n");

    // Export 2
    //   From selected layers 0, Line, and Arc, export
    //     object data, block attributes, link templates
    //   Erase drawing objects after export

    bExport = pExport->ExportObjectsAll(
        "0,Line,Arc",
        AcMapOracleExport::kBlockAttributes |
        AcMapOracleExport::kObjectData      |
        AcMapOracleExport::kLinkTemplates  |
        AcMapOracleExport::kEraseExported );

    if(bExport)
        acutPrintf("\nExport 2 successful!!\n");
}
```

```

else
    acutPrintf("\nExport 2 failed!\n");

// Export 3
// From all layers export all objects, but not
// object data, block attributes, link templates
// With erased objects, delete corresponding record
// Erase drawing objects after export

bExport = pExport->ExportObjectsAll(
    "",
    AcMapPracleExport::kUpdateErased |
    AcMapPracleExport::kEraseExported );

if(bExport)
    acutPrintf("\nExport 3 successful!!\n");
else
    acutPrintf("\nExport 3 failed!\n");

// Export 4
// Export objects in an ObjectIdArray

AcDbObjectIdArray idArray;

// idArray can be filled by manually
// selecting Objects from Model space
// or iterating thorough Model Space
// and selecting all

bExport = pExport->ExportObjects(
    idArray,
    AcMapPracleExport::kBlockAttributes |
    AcMapPracleExport::kObjectData |
    AcMapPracleExport::kLinkTemplates |
    AcMapPracleExport::kEraseExported );

if(bExport)
    acutPrintf("\nExport 4 successful!!\n");
else
    acutPrintf("\nExport 4 failed!\n");

// Export 5
// Insert new rows in database (objects are new)
// From all layers export all objects, and include
// object data, block attributes, link templates
// Erase drawing objects after export

bExport = pExport->ExportObjects(
    "",
    AcMapPracleExport::kDisregardIDs |
    AcMapPracleExport::kBlockAttributes |
    AcMapPracleExport::kObjectData |
    AcMapPracleExport::kLinkTemplates |
    AcMapPracleExport::kEraseExported );

if(bExport)

```

```
    acutPrintf("\nExport 5 successful!!\n");  
else  
    acutPrintf("\nExport 5 failed!\n");  
}
```

# Getting Corresponding IDs

The following sample demonstrates getting an entity's Oracle ID if its AutoCAD ID is known, and vice versa. It also demonstrates that round trips with corresponding IDs are reliable — that if you use an AutoCAD ID to get an Oracle ID, and then use the Oracle ID to get an AutoCAD ID, the two AutoCAD IDs are the same.

**Note** The sample assumes an open project drawing containing imported entities. It also assumes that the connection object is actually connected to a database. Click "Connecting to an Oracle Spatial Database" for sample code.

[Connecting to an Oracle Spatial Database \(sample\)](#) [Oracle Spatial and AutoCAD IDs \(overview\)](#)

[Getting corresponding IDs \(overview\)](#)

[Related samples](#)

```
#include "StdAfx.h"
#include "StdArx.h"
#include "AcMapOracleConnection.h"
#include "AcMapOracleIdentification.h"

BOOL Identification()
{
    // Get the connection object
    AcMapOracleConnection *pConnection = AcMapOracleGetConnection();

    // Create an identification object
    AcMapOracleIdentification *pId = new AcMapOracleIdentification(pConnection);

    AcDbObjectId AcadId;
    AcDbEntity *pEntity;
    ads_name en;
    ads_point pt;
    unsigned long ulOracleID;

    // Get an entity from the user
    if((acedEntSel("\nSelect an entity: ", en , pt) == RTNORM))
    {
        // Get the entity's AutoCAD ID
        acdbGetObjectID(AcadId, en);

        // Initialize the identification object with the AutoCAD ID
        bInit = pId->Init(AcadId);

        // Get the corresponding Oracle ID
        ulOracleID = pId->GetOracleID();

        char pBuf[100];

        // Get the entity's handle and print it to the AutoCAD Map text window
        if (Acad::eOk == acdbOpenAcDbEntity( pEntity, AcadId, AcDb::kForRead ))
        {
            AcadId.handle().getIntoAsciiBuffer(pBuf);
        }
    }
}
```

```
        acutPrintf("\nAcad handle sel:%s", pBuf);
        pEntity->close();
    }

    // Initialize the identification object with the Oracle ID
    bInit = pId->Init(ulOracleID);

    // Get the corresponding AutoCAD ID
    AcadId = pId->GetAcadID(
        acdbHostApplicationServices()->workingDatabase());

    // Get the entity's handle and print it to the AutoCAD Map text window
    if (Acad::eOk == acdbOpenAcDbEntity( pEntity, AcadId, AcDb::kForRead ))
    {
        AcadId.handle().getIntoAsciiBuffer(pBuf);
        acutPrintf("\nAcad handle sel:%s", pBuf);
        pEntity->close();
    }
}

acadSSFFree(en);

}
```



## AcMapProject:: CreatePropertyAlterationDefinition Function 1 of 2

[AcMapProject Class](#)

---

Allocates new `AcMapPropertyAlterationDefinition` object.

```
virtual AcMap::EErrCode  
CreatePropertyAlterationDefinition(  
    AcMapPropertyAlterationDefinition*& pAlteration) const;
```

Returns `AcMap::kOk` in case of success, error code otherwise.

`pAlteration`      Output new `AcMapPropertyAlteration` object. It is application's responsibility to erase it.

This object represents a stand alone property alteration. An application is responsible to delete the object allocated via this method.



## AcMapProject:: CreatePropertyAlterationDefinition Function 2 of 2

[AcMapProject Class](#)

---

Allocates new `AcMapPropertyAlteration` object.

```
virtual AcMap::EErrCode  
CreatePropertyAlterationDefinition(  
    AcMapPropertyAlteration*& pAlteration,  
    AcMap::EAlterationType kType) const;
```

Returns `AcMap::kOk` in case of success, error code otherwise.

**pAlteration** Output new `AcMapPropertyAlteration` object. It is application's responsibility to erase it.

**kType** Input the type of object allocated by this method, where  
    `AcMap::kAlterationTextEntity` = `AcMapTextAlteration` object  
    `AcMap::kAlterationHatch` = `AcMapHatchAlteration` object  
    `AcMap::kAlterationAnnotate` = `AcMapAnnotAlteration` object  
    Other `kType` arguments = `AcMapPropertyAlteration` object.

This object represents a stand alone property alteration. An application is responsible for removing the object allocated via this method calling `Erase()`.

**NOTE** `AcMap::kAlterationText` does not get you an `AcMapTextAlteration` object! The only correct `kType` for `AcMapTextAlteration` objects is `AcMap::kAlterationTextEntity`. When this method creates `AcMapTextAlteration` or `AcMapHatchAlteration` objects, it uses default values for their parameters.

Unlike similar methods, this method always allocates a new `pObj` object, which the application is responsible for deleting.



## AcMapAttachedDrawing:: ApplyThisQuery Function

[AcMapAttachedDrawing Class](#)

---

Executes the specified query against the attached drawing to select objects within the drawing.

```
virtual AcMap::EErrCode  
ApplyThisQuery(  
    AcDbObjectIdArray& aObjectSelection,  
    AcMapQuery* pQuery) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

**aObjectSelection**      Output IDs of the objects in the drawing that match the query. These IDs have valid values until the drawing has a heavy lock in place.

**pQuery**                  Input query definition.



## AcMapExpression:: TopologyExecute Function

[AcMapExpression Class](#)

---

Evaluates the expression against the specified topology element.

```
virtual AcMap::EErrCode  
TopologyExecute(  
    AcMapValue& tValue,  
    const char* pcTopologyName,  
    long topoElementId,  
    Adesk::Boolean fromCurDwg) = 0;
```

Returns nothing.

<code>tValue</code>	Output expression value.
<code>pcTopologyName</code>	Input topology name of an object to modify.
<code>topoElementId</code>	Input topology element ID of an entity to modify.
<code>fromCurDwg</code>	Input true if expression has to be executed against a topology in the current drawing or false if against a topology in a source drawing.

Topology must be loaded. If the Val type is not initialized, the type of the value is set automatically.



## AcMapQuery:: Execute Function 1 of 3

[AcMapQuery Class](#)

---

Executes the query against the given drawing set.

```
virtual AcMap::EErrCode  
Execute(  
    AcDbObjectIdArray& resultIds,  
    const AcMapDrawingSet* kpDwgSet) = 0;
```

Returns AcMap::kOk in case of success, AcMap::kErrUsrBreak if the user cancels the process, or another error code.

**resultIds**      Output selection of entities in the project drawing that meet the query conditions.

**kpDwgSet**      Input the drawing set pointer.

Objects that meet the query conditions are cloned from the source drawings in the drawing set to their associated project drawing. The result is a selection of objects in the project drawing. If an object has already been queried, it is not cloned. It is just added to the selection.



## AcMapQuery:: Execute Function 2 of 3

[AcMapQuery Class](#)

---

Executes the query against the drawing set of the given project.

```
virtual AcMap::EErrCode  
Execute(  
    AcDbObjectIdArray& resultIds,  
    const AcMapProject* kpProject) = 0;
```

Returns AcMap::kOk in case of success, AcMap::kErrUsrBreak if the user cancels the process, or another error code.

**resultIds**      Output selection of entities in the project drawing that meet the query conditions.

**kpProject**      Input the project pointer.

The result is a selection of objects in the project drawing.



## AcMapQuery:: Execute Function 3 of 3

[AcMapQuery Class](#)

---

Executes query against the given topology.

```
virtual AcMap::EErrCode  
Execute(  
    AcDbObjectIdArray& resultIds,  
    AcArray<AcDbObjectIdArray* >& resultEntityIdsPerTopoElement,  
    AcArray<long>& resultTopoElementIds,  
    const char* pszTopoName) = 0;
```

Returns AcMap::kOk in case of success, AcMap::kErrUsrBreak if the user cancels the process, or another error code.

<code>resultIds</code>	Output selection of entities in the current drawing that meet the query conditions.
<code>resultEntityIdsPerTopoElement</code>	Output array of *AcDbObjectIdArrays for each member of resultTopoElementIds. These are parallel arrays. The AcDbObjectArray pointers are allocated by this object and will be cleaned up in the destructor.
<code>resultTopoElementIds</code>	Output selection of topology elements in the current drawing that meet the query conditions.
<code>pszTopoName</code>	Input the topology name. The topology has to be loaded.

The result is a selection of objects in the current drawing. If an object has already been queried, it is not cloned. It is just added to the selection.



## AcMapQuery::FileIn Function

[AcMapQuery Class](#)

---

Restores the query definition from the result buffer.

```
virtual AcMap::EErrCode  
FileIn(  
    const struct resbuf* pResBuffer) = 0;
```

Returns AcMap::kOk in case of success, appropriate error code otherwise.

**pResBuffer**      Input result buffer.



## AcMapQuery::FileOut Function

[AcMapQuery Class](#)

---

Stores the query definition in a result buffer.

```
virtual AcMap::EErrorCode  
FileOut(  
    struct resbuf*& pResBuffer) const = 0;
```

Returns AcMap::kOk in case of success, appropriate error code otherwise.

**pResBuffer**      Output result buffer containing the query definition. Your application is responsible for deleting the result buffer. Use `acutRelRb()`.



## AcMapQuery:: GetStringToDisplay Function

[AcMapQuery Class](#)

---

Retrieves the string representation of the query.

```
virtual AcMap::EErrCode  
GetStringToDisplay(  
    char*& pszString) const = 0;
```

Returns AcMap::kOk in case of success, appropriate error code otherwise.

**pszString**      Output query string. Your application is responsible for releasing the memory allocated for it.



## AcMapQuery:: GetTopologyName Function

[AcMapQuery Class](#)

---

Retrieves topology name from a query definition.

```
virtual AcMap::EErrCode  
GetTopologyName(  
    const char*& kpszTopoName) const = 0;
```

Returns AcMap::kOk in case of success, AcMap::kErrObjectNotFound if query is not a topology query, or another error code.

**kpszTopoName**      Output the topology name if any has been assigned. Otherwise NULL.



## AcMapQuery:: IsDefined Function

[AcMapQuery Class](#)

---

Checks if any query conditions have been defined.

virtual bool

IsDefined();

Returns true if the query has at least one condition.



## AcMapQuery:: Select Function

[AcMapQuery Class](#)

---

Executes the query against the given topology.

```
virtual AcMap::EErrCode  
Select(  
    AcArray<long>& resultTopoElementIds,  
    const char* pszTopoName) = 0;
```

Returns AcMap::kOk in case of success, AcMap::kErrUsrBreak if the user cancels the process, or another error code.

<code>resultTopoElementIds</code>	Output selection of topology elements that meet the query conditions: polygons, links, or nodes, depending on the topology type.
<code>pszTopoName</code>	Input the topology name. The topology has to be loaded.

This function selects topological elements only. There is no cloning.



## AcMapQuery:: SetTopologyName Function

[AcMapQuery Class](#)

---

Sets topology name for a query definition.

```
virtual AcMap::EErrCode  
SetTopologyName(  
    const char* kpszTopoName) = 0;
```

Returns AcMap::kOk in case of success, appropriate error code otherwise.

**kpszTopoName**      Input the topology name.

The specified topology has to exist.



## AcMapPropertyAlterationDefinition:: Apply Function 1 of 2

[AcMapPropertyAlterationDefinition Class](#)

---

Applies property alteration to the entity specified by the ID.

```
virtual AcMap::EErrCode  
Apply(  
    const AcDbObjectId& Id) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

**Id**     Input object Id of an entity to modify.



## AcMapPropertyAlterationDefinition:: Apply Function 2 of 2

[AcMapPropertyAlterationDefinition Class](#)

---

Applies property alteration to the entity specified by the ID.

```
virtual AcMap::EErrCode  
Apply(  
    const char* topologyName,  
    long topoElementId) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

**topologyName**      Input name of the topology containing the entity to modify.

**topoElementId**      Input topology element ID of the entity to modify.



## AcMapPropertyAlterationDefinition:: Erase Function

[AcMapPropertyAlterationDefinition Class](#)

---

Destroys the property alteration data.

```
virtual AcMap::EErrCode  
Erase();
```

Returns AcMap::kOk in case of success, error code otherwise.

An application is still responsible for calling delete for the erased object to free the memory.



## AcMapPropertyAlterationDefinition:: ObjectId Function

[AcMapPropertyAlterationDefinition Class](#)

---

Accesses the ID of the associated internal object.

```
virtual AcMapId  
ObjectId();
```

Returns the ID.



## AcMapHatchAlteration:: Apply Function 1 of 2

[AcMapHatchAlteration Class](#)

---

Applies property alteration to the entity specified by the ID.

```
virtual AcMap::EErrCode  
Apply(  
    AcDbObjectIdArray& createdObjects,  
    const AcDbObjectId& Id) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

**createdObjects**      Output IDs of the objects created. No objects get created in the case of a simple entity alteration.

**Id**                    Input object ID of the entity to modify.



## AcMapHatchAlteration:: Apply Function 2 of 2

[AcMapHatchAlteration Class](#)

---

Applies property alteration to the entity specified by the ID.

```
virtual AcMap::EErrCode  
Apply(  
    AcDbObjectIdArray& createdObjects,  
    const char* topologyName,  
    long topoElementId) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

<code>createdObjects</code>	Output IDs of the objects created. No objects get created in the case of a simple entity alteration.
<code>topologyName</code>	Input topology name of the entity to modify.
<code>topoElementId</code>	Input topology element ID of the entity to modify.



## AcMapHatchAlteration:: Erase Function

[AcMapHatchAlteration Class](#)

---

Destroys the Property Alteration data.

```
virtual AcMap::EErrCode  
Erase();
```

Returns AcMap::kOk in case of success, error code otherwise.

An application is still responsible for calling delete for the erased object to free the memory.



## AcMapHatchAlteration:: GetExcludedObjects Function

[AcMapHatchAlteration Class](#)

---

Gets an array of objects to exclude from the hatch.

```
virtual AcMap::EErrCode  
GetExcludedObjects(  
    AcDbObjectIdArray& excludedObjects) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

`excludedObjects`      Output Ids of the objects to be excluded from this hatch.



## AcMapHatchAlteration:: SetExcludedObjects Function

[AcMapHatchAlteration Class](#)

---

Sets an array of objects to exclude from the hatch.

```
virtual AcMap::EErrCode  
SetExcludedObjects(  
    AcDbObjectIdArray& excludedObjects) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

**excludedObjects**      Input IDs of the objects to be excluded from this hatch.





## AcMapTextAlteration:: Apply Function 2 of 2

[AcMapTextAlteration Class](#)

---

Applies property alteration to the entity specified via the Id.

```
virtual AcMap::EErrCode  
Apply(  
    AcDbObjectIdArray& createdObjects,  
    const char* topologyName,  
    long topoElementId) = 0;
```

Returns AcMap::kOk in case of success, error code otherwise.

<code>createdObjects</code>	Output Ids of the objects created. No objects get created in the case of a simple entity alteration.
<code>topologyName</code>	Input topology name of an object to modify.
<code>topoElementId</code>	Input topology element Id of an entity to modify.



## AcMapTextAlteration:: Erase Function

[AcMapTextAlteration Class](#)

---

Destroys the Property Alteration data.

```
virtual AcMap::EErrCode  
Erase();
```

Returns AcMap::kOk in case of success, error code otherwise.

An application is still responsible for calling delete for the erased object to free the memory.

# Subclassing Custom Reactors

The following sample code illustrates subclassing custom reactors from AcMapOraclexxReactor classes. Classes and functions are declared, but function definitions, which would be highly implementation specific, are not shown. This is an example of what MyMapOracleReactors.h, one of the include files in the "Connecting to an Oracle Spatial Database" sample, would contain.

[Managing reactors \(overview\)](#) [Connecting to an Oracle Spatial Database \(sample\)](#)  
[Other related samples](#)

```
#include "AcMapOracleReactor.h"

class MyConnectionReactor : public AcMapOracleConnectionReactor
{
public:
    MyConnectionReactor();
    virtual ~MyConnectionReactor();

    virtual void Connected();
    virtual void Disconnected();
    virtual void BeforeConnect();
    virtual void BeforeDisconnect();
};

class MyExportReactor : public AcMapOracleExportReactor
{
public:
    MyExportReactor();
    virtual ~MyExportReactor();

    virtual void BeforeObjectCached(AcDbEntity *);
    virtual void ObjectCached(AcDbEntity *, unsigned long);
    virtual void ObjectRejected(AcDbEntity *);
    virtual void BeforeObjectsExported(std::vector &);
    virtual void ObjectsExported(std::vector &);
};

class MyImportReactor : public AcMapOracleImportReactor
{
public:
    MyImportReactor();
    virtual ~MyImportReactor();

    virtual void BeforeRecordImport(const ODynaset &);
};
```

```
virtual void RecordImported(const ODynaset &,AcDbEntity *);  
virtual void RecordRejected(const ODynaset &);  
  
};
```