Object Model

Click on an object to view the description.

Modified Event

Triggered when an object or collection in the drawing has been modified.

See Also | Example

Signature

object.Modified()

Object

All Land objects can trigger the Modified Event An object expression that evaluates to a valid container object.

Remarks

This event will be triggered whenever the object is modified. Modification includes whenever the value of a property is set, even if the new value is equal to the current value.

When coding in VBA, you must provide an event handler for all objects enabled for the Modified event. If you do not provide a handler, VBA may terminate unexpectedly.

No events will be fired while a modal dialog is being displayed.

Refer to the <u>Land ActiveX and VBA Developer's Guide</u> for a list of support events.

Methods



Α

Add Creates a member object and adds it to the appropriate collection.

AddCurve Appends a curve to the alignment or parcel.

AddLabelAt Adds a contour label at the specified location.

AddLine Appends a line to a parcel.

AddSpiral Appends a spiral to the alignment.

AddTangent Appends a tangent to the alignment.

AddToAllElevations Adds a fixed amount to all elevations in the Surface.

<u>AlignmentFromObjectID</u> Given the object ID for an entity, returns the Alignment name.

<u>AreaVolume</u> Given an array of points, returns the cut, fill, and net volume.

<u>ArrayToPointString</u> Given an array of point numbers, returns a string with all of the point numbers.

В

Build The Build method will recalculate the Surface.

C

<u>ClearOverides</u> Clears all overrides for the project database point group.

Composite Creates a composite surface from two existing surfaces.

<u>ConvertToCurrentAreaDisplay</u> Converts an area value using the LinearUnit and AreaDisplayUnit settings.

<u>ConvertToCurrentVolumeDisplay</u> Converts a volume value using the LinearUnit and VolumeDisplayUnit settings.

<u>Copy</u> Copies a Prototype or Surface.

D

Delete Deletes a specified object.

<u>DifferenceGrid</u> Creates a surface from two existing surfaces using the Grid method.

<u>DoubleToStaFormat</u> Given a double, returns a string with the Station Format applied.

Ε

EastNorthToXy Given an Easting and Northing, will return the AutoCAD XY

<u>ElevationAt</u> (civil engineering feature) Given a station, returns the elevation for a profile.

<u>ExternalStaToInternal</u> Given an external Station value (one using Station Equations), returns the internal Station value.

F

FindAllConnectingEdges Returns all edges for a given Surface point.

FindAllFaces Redefines the collection with all faces on the surface.

FindConnectingEdge Given a Surface point, retrieves the closest edge.

FindFace Given an Easting / Northing coordinate, returns the Face.

<u>FindPath</u> Give two coordinates, reads in all of the Faces that lie on that line.

<u>FindPoint</u> Given an Easting / Northing coordinate, finds the closest Surface.

<u>GetBoundingBox</u> Gets the bounding box for the surface.

<u>GetDouble</u> Gets the specified preference setting.

GetElevation Given a Northing and Easting, return the elevation.

GetInteger Gets the specified preference setting.

<u>GetLayerName</u> (civil engineering feature) Returns the layer names of the alignment cross section or alignment profile.

<u>GetStaStrWithEquations</u> Given an internal station value, returns that station as a formatted string with Station Equations applied.

<u>GetStaWithEquations</u> Given an internal station value, returns that station value with Station Equations applied.

GetString Gets the specified preference setting.

Н

ı

<u>Import</u> Draws the alignment, cross section, existing ground profile, parcel, or surface into the current drawing.

<u>InstantGrade</u> (civil engineering feature) Given a station, returns the instantaneous grade and the algebraic difference.

<u>Item</u> Gets the member object at a given index in a collection.

J

Κ

1

<u>LineIntersection</u> Returns the intersection points of a line with the Alignment.

<u>LoadSetupProfile</u> Load the specified drawing setup profile.

<u>LockPoints</u> Locks a list of point numbers.

M

Ν

NewProjectBased Creates a new AutoCAD Land Desktop document.

0

OffsetElevationToXy (civil engineering feature) Given an offset and elevation, return the AutoCAD XY coordinates.

Open Opens an existing AutoCAD Land Desktop drawing or surface.

<u>OpenProjectBased</u> Opens an existing AutoCAD Land Desktop drawing file (DWG).

P

<u>Paste</u> The Paste method will take another Surface an paste it onto the given Surface.

<u>PerpIntersection</u> Returns the perpendicular intersection of a point with an Alignment.

<u>PointByNumber</u> Returns a point in the project database by number.

<u>PointCodeDescription</u> (civil engineering feature) Returns the description of a cross section point code.

<u>PointLocation</u> Given a station and offset, returns the Northings and Eastings.

<u>PointNumberFromObjID</u> Given a AutoCAD object ID, PointNumberFromObjID will return the point number.

<u>PointStringToArray</u> Given a point string, returns an array with all of the point numbers.

<u>ProfileByType</u> (civil engineering feature) Returns the Existing or Finished Ground Profile by type.

Q

R

Remove All Remove all method for Alignments and StationEquations.

RemoveAllLabels Removes all contour labels.

RemoveLabelAt Removes the contour label closest to the specified location.

Rename Renames a surface.

S

<u>SampleElevations</u> Given a start and end point, returns an array of derived points from the Surface.

Save Writes objects to specific databases.

SaveAsDefault Saves the DatabasePreferences to the registry.

<u>SaveSetupProfile</u> Saves the current drawing settings to a drawing setup profile.

<u>SectionByStation</u> (civil engineering feature) Returns a Cross Section object by specifing its station.

<u>SectionVolume</u> (civil engineering feature) Calculates the Cross Section volumes.

<u>SetBoundingBox</u> Sets the bounding box for the surface.

<u>SetDouble</u> Sets the specified preference setting.

SetInteger Sets the specified preference setting.

SetReferenceCurve Sets the reference curve for the Curve Text.

<u>SetString</u> Sets the specified preference setting.

<u>StationElevationToXy</u> (civil engineering feature) Given an alignment cross section or alignment profile station and elevation, return the AutoCAD XY coordinates.

<u>StationOffset</u> Given a Northings and Eastings, returns the station, and an offset, and the polar direction.

T

U

<u>UnlockPoints</u> Unlocks a list of point numbers.

V

W

X

XyToEastNorth Given a AutoCAD X, Y, returns the Easting and Northing.

<u>XyToOffsetElevation</u> (civil engineering feature) Given an AutoCAD XY cooridnates, return the cross section offset and elevation.

<u>XyToStationElevation</u> (civil engineering feature) Given an AutoCAD XY cooridnates, return the profile station and elevation.

Υ

Ζ

Add Method

Creates a member object and adds it to the appropriate collection.

See Also | Example

Signature: Overview

- 1 Alignments
- 1 **Boundaries**
- l BreakLines
- l CogoPoints
- l ContourItems
- 1 **DEMFiles**
- l <u>DescriptionKeyFile</u>
- l <u>DescriptionKeyFiles</u>
- l <u>EGProfiles</u> (Civil Engineering Feature)
- 1 FGProfiles (Civil Engineering Feature)
- 1 Parcels
- l PointFiles
- l <u>PointGroupNames</u>

- 1 PointGroups
- 1 Projects
- 1 PVIs (Civil Engineering Feature)
- 1 StationEquations
- 1 Surfaces

Creates a new Alignment and adds it to the Alignments collection.

Signature

RetVal = object.Add(Name, StartingStation)

object

Alignments The object or objects this property

applies to.

Name

String; input-only

New alignment name.

Starting Station

Double; input-only The starting station.

RetVal

Alignment object

The newly added Alignment object.

Remarks

The new alignment must be saved for any changes to take affect. You can add a new alignment to the collection and then define the underlying geometry. Be advised that if you save an alignment without later adding any tangents, curves or spirals, you will not be able to edit the geometry using the alignment editor.

Adds a Boundary to the Surface.

Signature

RetVal = object.Add(Type, bIsBreakLine, Coordinates[, Description])

object

Boundaries

The object or objects this property applies to.

Type

eAeccBoundaryType enum; input-only

kBoundaryTypeShow: Boundary is visible

kBoundaryTypeHide: Boundary is hidden

kBoundaryTypeOuter:

Boundary is the outer most

boundary

kBoundaryTypeUninit: Boundary in not initialized.

bIsBreakLine

Boolean; input only

TRUE: Boundary is a breakline FALSE: Boundary is not a breakline

Coordinates

Variant (array of doubles); input-only

The points in Easting, Northing, Elevation format.

Description

String; input only; optional

The description of the Boundary.

RetVal

Boundary object

The newly added Boundary object.

Remarks

Outer boundaries are always visible and must be the first boundary defined.

Creates a new BreakLine for the Surface.

Signature

RetVal = object.Add(Coordinates[, Description])

object

BreakLines

The object or objects this property applies to.

Coordinates

Variant (array of doubles); input-only

The points in Easting, Northing, Elevation format.

Description

String; input-only; optional

The description for the BreakLine.

RetVal

BreakLine object

The newly added BreakLine object.

Remarks

Add creates the BreakLine by adding it to the Surface. It will also add the BreakLine to the <u>BreakLines</u> collection. Only a Standard BreakLine can be added. The Surface must be built for the change to take affect.

Creates a new CogoPoint and adds it to the collection for the project

database.

Signature

RetVal = object.Add(Coordinates, Format)

object

CogoPoints

The object or objects this property applies to.

Coordinates

Variant (2 or 3 element array of doubles); input-only. If the array has 2 elements, no elevation will be

assigned.

Double[0]: Easting, or X based on PointStyle

Double[1]: Northing, or Y based on PointStyle

Double[2]: Optional: Elevation or

kPointNoElevation

Format

eAeccCoordinateFormat enum; input-only

kCoordinateFormatENZ: Point is in Easting,

Northing, elevation

Point is in X, Y,

kCoordinateFormatXYZ: elevation. AutoCAD

coordinates

RetVal

CogoPoint object

The newly added <u>CogoPoint</u> object.

Adds a ContourItem to the Surface definition.

Signature

RetVal = object.Add(Coordinates)

object

ContourItems

The object or objects this property applies to.

Coordinates

Variant (array of doubles); input-only

An array of doubles representing Easting, Northing,

Elevation.

RetVal

ContourItem object

The newly added **ContourItem** object.

Adds a Digital Elevation Model file to the Surface definition.

Signature

RetVal = object.Add(FullName, [CoordinateZone])

object

DEMFiles

The object or objects this property applies to.

FullName

String; input-only

Full name of the DEM file to be added.

CoordinateZone

String; input-only

The coordinate system syntax for the zone.

RetVal

DEMFile object

The newly added **DEMFile** object.

Creates a new DescriptionKey.

Signature

RetVal = object.Add(syntax)

object

DescriptionKeyFile

The object or objects this property applies to.

syntax

String; input-only

The syntax of the new **DescriptionKey**

RetVal

DescriptionKey object

The newly added **DescriptionKey** object.

Creates a new DescriptionKeyFiles and adds it to the collection for the project database.

Signature

RetVal = object.Add(Name)

object

DescriptionKeyFiles

The object or objects this property applies to.

Name

String; input-only

The name of the new DescriptionKeyFiles

RetVal

<u>DescriptionKeyFile</u> object The newly added <u>DescriptionKeyFile</u> object.

Creates a new EGProfile and adds it to the EGProfiles collection.

Signiture

RetVal = Object.Add(Type, SurfaceName)

object

EGProfiles

The object or objects this property applies to.

Type

eAeccEGProfilesType enum; input-only

The type for the new EGProfile.

kEgCenter: Existing ground center profile

kEgLeft: Existing ground left profile

kEgNone: Existing ground none

kEgRight: Existing ground right profile

String

String; input-only

The name of the surface.

RetVal

EGProfile object

The newly added EGProfile object.

Adds a Finished Grade Profile to the project

Signiture

RetVal = Object.Add(Type)

object

FGProfiles

The object or objects this property applies to.

Type

eAeccFGProfileType enum; input-only The type of finished grade profile to add.

kFgCenter: Finished grade center profile

kFgDitchLeft: Finished grade left ditch profile

kFgDitchRight: Finished grade right ditch profile

kFgLeft1: Finished grade left 1 profile

kFgLeft2: Finished grade left 2 profile

kFgLeft3: Finished grade left 3 profile

kFgLeft4: Finished grade left 4 profile

kFgLeft5: Finished grade left 5 profile

kFgLeft6: Finished grade left 6 profile

kFgLeft7: Finished grade left 7 profile

kFgLeft8: Finished grade left 8 profile

kFgNone: Finished grade none

kFgRight1: Finished grade right 1 profile

kFgRight2: Finished grade right 2 profile

kFgRight3: Finished grade right 3 profile

kFgRight4: Finished grade right 4 profile

kFgRight5: Finished grade right 5 profile

kFgRight6: Finished grade right 6 profile

kFgRight7: Finished grade right 7 profile

kFgRight8: Finished grade right 8 profile

RetVal

FGProfile object

The newly added **FGProfile** object.

Remarks

A Finished Grade Profile can not start or end with a Vertical Curve.

Creates a new Parcel and adds it to the ExistingParcelAligns collection.

Signiture

RetVal = Object.Add(Name)

object

Parcels

The object or objects this property applies to.

Name

String; input-only

The name of the parcel.

RetVal

Parcel object

The newly added **Parcels** object.

Adds a point file to the PointFiles collection.

Signature

RetVal = object.Add(FullName)

object

PointFiles

The object or objects this property applies to.

FullName

String; input-only

Full name of the point file to be added.

RetVal

PointFile object

The newly added **PointFile** object.

Remarks

PointFile.Add does not check for the existence of the file. The new PointFile will use the default file format of Point Number, Easting, Northing, Elevation (space-delimited). Each field in the point file must be seperated with a single space.

Adds a new PointGroupName to the PointGroupNames collection.

Signature

RetVal = object.Add(Name)

object

PointGroupNames

The object or objects this property applies to.

Name

String; input-only

Name of the PointGroupName to add.

RetVal

PointGroupName object

The newly added **PointGroupName** object.

Add a new point group to the project database.

Signature

RetVal = object.Add(Name, PointList)

object

PointGroups

The object or objects this property applies to.

GroupName

String; input-only

The new point group name.

PointList

String; input-only The point list.

RetVal

PointGroup object

The newly added PointGroup object.

Creates a new project.

Signature

RetVal = object.Add(ProjectName, PrototypeName[, CreatePDb])

object

Projects

The object or objects this property applies to.

ProjectName

String; input-only

The name of the project to delete.

PrototypeName

String; input-only

The name of the prototype to use for the project.

CreatePDb

Boolean; input-only; optional

True: The Cogo Point database will be created

(default).

False: The Cogo Point database will not be created

RetVal

Project object

The newly added **Project** object.

Remarks

The new project is created at the <u>ProjectPath</u> directory. The default subfolder to save drawings in is the \dwg subfolder.

If CreatePDb is True, the Cogo Point database will be created. The Description field size will be 32 characters. Point Names will be enable with a size of 16 characters. The default value for CreatePDb is True.

Adds a PVI.

Signiture

RetVal = Object.Add(Station, Elevation, Curvelength)

object

PVIs

The object or objects this property applies to.

Station

Double; input-only

The Station for the new PVI.

Elevation

Double; input-only

The Elevation for the new PVI.

CurveLength

Double; input-only

The CurveLength for the new PVI.

RetVal

PVI object

The newly added **PVI** object.

Adds a station equation into the alignment.

Signature

RetVal = object.Add(StationBack, StationAhead, Type)

object

StationEquations

The object or objects this property applies to.

StationBack

Double; input-only

The back station for the station equation.

StationAhead

Double; input-only

The ahead station for the station equation.

Type

eAeccStationEquationType enum; input-only

kIncreasing: Station numbers increase along the

alignment

kDecreasing: Station numbers decrease along

the alignment

RetVal

StationEquation object

The newly added <u>StationEquation</u> object.

Creates a new Surface and adds it to the Surfaces collection.

Signature

RetVal object.Add(SurfaceName)

object

<u>Surfaces</u>

The object or objects this property applies to.

SurfaceName

String; input-only New surface name.

RetVal

Surface object

The newly added <u>Surface</u> object.

Remarks

If the name of the surface is blank, then a temporary surface will be created. A temporary surface is useful for generating a surface in an AutoCAD drawing. A temporary surface can not be saved.

AddCurve Method

Appends a curve to the alignment or parcel.

See Also | Example

Signature: Overview

1 Alignment

1 Parcel

Appends a curve to the alignment.

Signature

RetVal = object.AddCurve(StartEasting, StartNorthing, CenterEasting, CenterNorthing, EndEasting, EndNorthing, bCCWFlag)

object

Alignment The object or objects this property

applies to.

StartEasting

Double; input-only

The Easting coordinate for the beginning of the

curve.

StartNorthing

Double; input-only

The Northing coordinate for the beginning of the

curve.

CenterEasting

Double; input-only

The Easting coordinate for the center of the curve.

CenterNorthing

Double; input-only

The Northing coordinate for the center of the curve.

EndEasting

Double; input-only

The Easting coordinate for the end of the curve.

EndNorthing

Double; input-only

The Northing coordinate for the end of the curve.

bCCWFlag

Boolean; input-only

TRUE: If the curve is drawing in a counterclockwise

direction.

FALSE: If the curve is drawing in a clockwise

direction.

RetVal

AlignCurve object

The Newly added AlignCurve object

Remarks

A check will be made to see if the beginning of the new curve is equal to the end of the previous entity. If the two are not equal, an error (AECC_E_ALN_INVALID_ENDPOINT) will be generated and the curve will not be added.

No check will be made to see if the new curve is tangent to the previous entity.

Appends a curve to the parcel.

Signature

RetVal = object.AddCurve(StartEasting, StartNorthing, CenterEasting, CenterNorthing, EndEasting, EndNorthing, bCCWFlag)

object

<u>Parcel</u>

The object or objects this property applies to.

StartEasting

Double; input-only

The Easting coordinate for the beginning of the

curve.

StartNorthing

Double; input-only

The Northing coordinate for the beginning of the

curve.

CenterEasting

Double; input-only

The Easting coordinate for the center of the curve.

CenterNorthing

Double; input-only

The Northing coordinate for the center of the curve.

EndEasting

Double; input-only

The Easting coordinate for the end of the curve.

EndNorthing

Double; input-only

The Northing coordinate for the end of the curve.

bCCWFlag

Boolean; input-only

TRUE: If the curve is drawing in a counterclockwise

direction.

FALSE: If the curve is drawing in a clockwise

direction.

RetVal

ParcelCurve object

The Newly added ParcelCurve object.

AddLabelAt Method

Adds a contour label at the specified location.

See Also | Example

Signature

object.AddLabelAt(X, Y)

object

AeccContour The object or objects this property

applies to.

X

Double; input-only

The X coordinate of the new label.

Y

Double; input-only

The Y coordinate of the new label.

Remarks

The label will be placed at the closest location on the contour based on the X, Y point.

AddLine Method

Appends a line to the parcel.

See Also | Example

Signature

RetVal = object.AddParcel(StartEasting, StartNorthing, EndEasting, EndNorthing)

object

Parcel The object or objects this property applies to.

StartEasting

Double; input-only

The Easting coordinate for the beginning of the line.

StartNorthing

Double; input-only

The Northing coordinate for the beginning of the

line.

EndEasting

Double; input-only

The Easting coordinate for the end of the line.

EndNorthing

Double; input-only

The Northing coordinate for the end of the line.

RetVal

ParcelLine object
The newly added ParcelLine object.

AddSpiral Method

Appends a spiral to the alignment.

See Also | Example

Signature

RetVal = object.AddSpiral(StartEasting, StartNorthing, PiEasting, PiNorthing, EndEasting, EndNorthing, LExt, LOffset, SpiralType1)

object

Alignment The object or objects this property

applies to.

StartEasting

Double; input-only

The Easting coordinate for the beginning of the

spiral.

StartNorthing

Double; input-only

The Northing coordinate for the beginning of the

spiral.

PiEasting

Double; input-only

The Easting coordinate for the PI of the spiral.

PiNorthing

Double; input-only

The Northing coordinate for the PI of the spiral.

EndEasting

Double; input-only

The Easting coordinate for the end of the spiral.

EndNorthing

Double; input-only

The Northing coordinate for the end of the spiral.

LExt

Double; input-only

The external length of the spiral. 0.0 for a simple

spiral.

LOffset

Double; input-only

The offset distance. 0.0 for non-offset spirals.

SpiralType1

eAeccSpiralType enum; input-only

kClothoid: Clothoid spiral

kSinusoid: Sinusoid spiral

kCosinusoid: Cosinusoid spiral

kQuadratic: Quadratic spiral

RetVal

AlignSpiral object

The newly added AlignSpiral object.

Remarks

A check will be made to see if the beginning of the new spiral is equal to

the end of the previous entity. If the two are not equal, an error (AECC_E_ALN_INVALID_ENDPOINT) will be generated and the spiral will not be added.

No check will be made to see if the new spiral is tangent to the previous entity.

Lext represents the length from the TS to the External Point in a compound spiral condition.

AddTangent Method

Appends a tangent to the alignment.

See Also | Example

Signature

RetVal = object.AddTangent(StartEasting, StartNorthing, EndEasting, EndNorthing)

object

Alignment The object or objects this property

applies to.

StartEasting

Double; input-only

The Easting coordinate for the beginning of the

tangent.

StartNorthing

Double; input-only

The Northing coordinate for the beginning of the

tangent.

EndEasting

Double; input-only

The Easting coordinate for the end of the tangent.

EndNorthing

Double; input-only

The Northing coordinate for the end of the tangent.

RetVal

AlignTangent object The newly added AlignTangent object.

Remarks

A check will be made to see if the beginning of the new tangent is equal to the end of the previous entity. If the two are not equal, an error (AECC_E_ALN_INVALID_ENDPOINT) will be generated and the tangent will not be added.

AddToAllElevations Method

Adds a fixed amount to all elevations in the Surface.

See Also | Example

Signature

object.AddToAllElevations(Delta)

object

Surface The object or objects this property applies

to.

Delta

Double; input-only

Elevation to be added / subtracted to all elevations.

Remarks

AddToAllElevations effectively raises or lowers the surface by DeltaElevation. The Surface will not be rebuilt but will be saved when AddToAllElevations is called.

Edit history must be turned on when the Surface is rebuilt for the changes to been seen.

AlignmentFromObjectID Method

Given the object ID for an entity, returns the Alignment name.

See Also | Example

Signature

RetVal = object.AlignmentFromObjectID(ObjectID)

object

Alignments The object this property applies to.

ObjectID

Long; read-only

The Object ID for the entity.

RetVal

Variant (array of strings)

An array of Alignment names.

AreaVolume Method

Given an array of points defining an area, returns the cut, fill, and net volume.

See Also | Example

Signature

object.AreaVolume(Tolerance, Coordinates, Cut, Fill, Net)

object

Surface The object or objects this property applies

to.

Tolerance

Double; input-only

Coordinates

Variant (array of doubles); input-only

A closed array of 3D points defining an area

Cut

Double; output-only The cut volume.

Fill

Double; output-only

The fill volume.

Net

Double; output-only

The net volume.

ArrayToPointString Method

Given an array of point numbers, returns a string with all of the point numbers.

See Also | Example

Signature

RetVal = object.PointStringToArray(Points)

object

CogoPoints The object or objects this property

applies to.

Points

Variant (array of longs); input-only Each element is a point number.

RetVal

String

Comma delimited string of points to be locked with

groups separated by hyphens.

Remarks

The returned point string will be condensed; if the array has point 1 through 10, then the point string returned is "1-10".

Build Method

Builds Watersheds or recalculates the Surface.

See Also | Example

Signature: Overview

1 Surface

1 WaterSheds

The Build method will recalculate the Surface.

Signature

object.Build()

object

<u>Surface</u> The object or objects this property applies to.

Remarks

When changes are made to a Surface, the Surface needs to be recalculated. The Build method will do the recalculation. The status of a surface will only be accurate when the time between a Build and an addition of Surface Input data is greater than the resolution of the file system.

Builds the WaterSheds for the Surface.

Signature

object.Build(MinDepressionDepth, MinDepressionArea, bExceedBoth)

object

WaterSheds

The object or objects this property applies to.

MinDepressionDepth

Double; input-only

The minimum depression depth for each watershed.

MinDepressionArea

Double; input-only

The minimum depression area for each watershed.

bExceedBoth

Boolean; input-only

TRUE: Both minimum depression conditions must

be met.

FALSE: Either minimum depression setting must be

met.

Remarks

Any change in the Surface statistics will not occur until the Surface and Surfaces objects are released and created again. For example, WaterSheds.Build may add TinPoints to the Surface. The NumberOfPoints for the Surface will not be updated until both the Surfaces collection and the Surface object are updated.

ClearOverrides Method

Clears all overrides for the project database point group.

See Also | Example

Signature

object.ClearOverrides()

object

PointGroup The object or objects this property applies to.

Composite Method

Creates a composite surface from two existing surfaces.

See Also | Example

Signature

RetVal = object.Composite(NewSurface, ParentSurface, ChildSurface)

object

Surfaces The object or objects this property applies

to.

NewSurface

String; input-only

The name of the new surface to be created.

ParentSurface

String; input-only

The name of the parent surface.

ChildSurface

String; input-only

The name of the child surface.

RetVal

Surface object

The newly added Surface object.

ConvertToCurrentAreaDisplay Method

Converts an area value using the LinearUnit and AreaDisplayUnit settings.

See Also | Example

Signature

RetVal = object.ConvertToCurrentAreaDisplay(Area)

object

DatabasePreferences The object or objects this

property applies to.

Area

Double; input-only The area to convert.

RetVal

Double

The resulting area.

ConvertToCurrentVolumeDisplay Method

Converts a volume value using the LinearUnit and VolumeDisplayUnit settings.

See Also | Example

Signature

RetVal = object.ConvertToCurrentVolumeDisplay(Volume)

object

DatabasePreferences The object or objects this

property applies to.

Volume

Double; input-only

The volume to convert.

RetVal

Double

The resulting volume.

Copy Method

Creates a copy of an existing object.

See Also | Example

Signature: Overview

l Prototypes

1 Surfaces

Copies a prototype.

Signature

RetVal = object.Copy(FromName, ToName)

object

Prototypes The object or objects this property

applies to.

FromName

String; input-only

The name of the prototype you wish to copy.

ToName

String; input-only

The name of the copy.

RetVal

Prototype object

The newly added Prototype object.

Remarks

The new prototype is created at the **PrototypePath** directory.

Copies a surface.

Signature

RetVal = object.Copy(SurfaceName)

object

Surfaces

The object or objects this property applies to.

SurfaceName

String; input-only

The name of the surface to be copied.

RetVal

Surface object

The newly added Surface object.

Remarks

The new surface created will have "Copy of" prefixed to the new surface name.

Delete Method

Deletes a specified object.

See Also | Example

Signature: Overview

- 1 Alignments
- 1 **Boundaries**
- 1 BreakLines
- l CogoPoints
- l ContourItems
- 1 **DEMFiles**
- l <u>DescriptionKeyFile</u>
- l <u>DescriptionKeyFiles</u>
- l <u>EGProfiles</u> (Civil Engineering Feature)
- 1 FGProfiles (Civil Engineering Feature)
- 1 Parcels
- l PointFiles
- l <u>PointGroupNames</u>

- l PointGroups
- 1 Projects
- l Prototypes
- 1 PVIs (Civil Engineering Feature)
- 1 StationEquations
- 1 Surfaces

Deletes an Alignment and removes it from the Alignments collection.

Signature

object.Delete(Name)

object

Alignments The object or objects this property

applies to.

Name

String; input-only

The alignment name to delete.

Remarks

Delete will remove the alignment from the database and the drawing. If the deleted alignment was set current there will not be any alignment set current after the delete has taken place.

Given a Boundary identifier, deletes that Boundary from the Surface.

Signature

object.Delete(Id)

object

Boundaries

The object or objects this property applies to.

Id

Long; input-only

The identifier of the boundary.

Deletes the BreakLine from the Surface.

Signature

object.Delete(Id)

object

BreakLines

The object or objects this property applies to.

Id

Long; input-only

The Id of the breakline to delete.

Remarks

Delete removes the BreakLine from the Surface. It will also remove the BreakLine from the <u>BreakLines</u> collection. The Surface must be built for the change to take affect.

Deletes the point from the project database.

Signature

object.Delete(PointNumber)

object

CogoPoints

The object or objects this property applies to.

PointNumber

Long; input-only

The Cogo Point number to be deleted.

Deletes a Digital Elevation Model file from the Surface.

Signature

object.Delete(FullName)

object

DEMFiles

The object or objects this property applies to.

FullName

String; input-only

Full name of the DEM file to be deleted.

Deletes a ContourItem from the Surface.

Signature

object.Delete(Id)

object

ContourItems

The object or objects this property applies to.

Id

Long; input-only

The Id of the ContourItem to be deleted.

Deletes the DescriptionKey.

Signature

object.Delete(syntax)

object

DescriptionKeyFile

The object or objects this property applies to.

syntax

String; input-only

The syntax of the **DescriptionKey** to delete.

Deletes the DescriptionKeyFile and removes it from the collection for the project database.

Signature

object.Delete(Name)

object

DescriptionKeyFiles

The object or objects this property applies to.

Name

String; input-only

The name of the <u>DescriptionKeyFile</u> to delete.

Deletes a EGProfile and removes it from the EGProfiles collection.

Signiture

Object.Delete(Type, SurfaceName)

object

EGProfiles

The object or objects this property applies to.

Type

eAeccEGProfilesType enum; input-only

The type to delete from the EGProfiles collection.

kEgCenter: Existing ground center profile

kEgLeft: Existing ground left profile

kEgRight: Existing ground right profile

SurfaceName

String; input-only

The name of the surface.

Deletes a Finised Ground Profile.

Signiture

Object.Delete(Type)

object

FGProfiles

The object or objects this property applies to.

Type

eAeccFGProfileType enum; input-only

The type of Finished Ground Profile to delete.

kFgCenter: Finished ground center profile

kFgDitchLeft: Finished ground left ditch profile

kFgDitchRight: Finished ground right ditch profile

kFgLeft1: Finished ground left profile 1

kFgLeft2: Finished ground left profile 2

kFgLeft3: Finished ground left profile 3

kFgLeft4: Finished ground left profile 4

kFgLeft5: Finished ground left profile 5

kFgLeft6: Finished ground left profile 6

kFgLeft7: Finished ground left profile 7

kFgLeft8: Finished ground left profile 8

kFgNone: Finished ground none

kFgRight1: Finished ground right profile 1

kFgRight2: Finished ground right profile 2

kFgRight3: Finished ground right profile 3

kFgRight4: Finished ground right profile 4

kFgRight5: Finished ground right profile 5

kFgRight6: Finished ground right profile 6

kFgRight7: Finished ground right profile 7

kFgRight8: Finished ground right profile 8

Deletes a Parcel from the Parcels collection.

Signature

object.Delete(Name)

object

Parcels

The object or objects this property applies to.

Name

String; input-only

The name of the Parcel to delete.

Deletes a point file from the PointFiles collection.

Signature

object.Delete(FullName)

object

PointFiles

The object or objects this property applies to.

FullName

String; input-only

The full name of the PointFile to delete.

Remarks

Remove will not delete the physical file; it only removes the FileName from the Surface definition.

Deletes a PointGroupName from the PointGroupNames collection.

Signature

object.Delete(Name)

object

PointGroupNames

The object or objects this property applies to.

Name

String; input-only

The name of the PointGroupName to delete.

Deletes a point group from the project database.

Signature

object.Delete(Name)

object

PointGroups

The object or objects this property applies to.

Name

String; input-only

Deletes a specific project

Signature

object.Delete(ProjectName)

object

Projects

The object or objects this property applies to.

ProjectName

String; input-only

The name of the project to delete.

Remarks

The Delete method deletes everything in the project name folder, including the drawing files if they are located in the project folder.

No warning dialog is presented. The project is deleted, and not moved to the Windows Recycle Bin.

Deletes the specified prototype.

Signature

object.Delete(PrototypeName)

object

Prototypes

The object or objects this property applies to.

PrototypeName

String; input-only

The name of the prototype you wish to delete.

Remarks

A warning dialog box is displayed, informing you that all files and folders within the prototype folder will be deleted.

Deletes a PVI.

Signiture

Object.Delete(Station)

object

PVIs

The object or objects this property applies to.

Station

Double; input-only

The Station of the PVI to delete.

Deletes a station equation from the alignment.

Signature

object.Delete(Index)

object

StationEquations

The object or objects this property applies to.

Index

Variant; input-only

The index location in the collection for the member item to query. The index must be either an integer or a string. If an integer, the index must be between 0 and N-1, where N is the number of objects in the

collection or selection set.

Remarks

If a station equation is deleted, then the indices for the <u>StationEquations</u> collection will change.

Deletes the Surface from the Project and from the Surfaces collection.

Signature

object.Delete(SurfaceName)

object

Surfaces

The object or objects this property applies to.

SurfaceName

String; input-only

Name of the surface to be deleted.

DifferenceGrid Method

Creates a surface from two existing surfaces using the Grid method.

See Also | Example

Signature

RetVal = object.DifferenceGrid(NewSurface, Surface1, Surface2, X, Y, Rows, Columns, OriginX, OriginY, Angle)

object

Surfaces The object or objects this property applies

to.

NewSurface

String; input-only

The name of the new surface to be created.

Surface1

String; input-only

The name of the first surface.

Surface2

String; input-only

The name of the second surface.

X

Double; input-only

The width of the grid cell.

Y

Double; input-only

The height of the grid cell.

Rows

Integer; input-only

The number of rows for the grid.

Columns

Integer; input-only

The number of columns for the grid.

OriginX

Double; input-only

The X coordinate of the grid origin.

Origin Y

Double; input-only

The Y coordinate of the grid origin.

Angle

Double; input-only

The rotation angle of the grid.

RetVal

Surface object

The newly added Surface object.

DoubleToStaFormat Method

Given a double, returns a string with the Station Format applied

See Also | Example

Signature

RetVal = object.DoubleToStaFormat(Station)

object

Alignments The object this property applies to.

Station

Double; read-only

The double to be formatted.

RetVal

String

Station formatted string.

Remarks

The Alignments DoubleToStaFormat method differs somewhat from the Alignment <u>GetStaStrWithEquations</u> method. DoubleToStaFormat does not use Station Equations and is not specific to an Alignment.

EastNorthToXy Method

Given an Easting and Northing, will return the AutoCAD XY

See Also | Example

Signature

RetVal = object.EastNorthToXy(EastNorth)

object

Utility The object or objects this property applies to.

EastNorth

Variant (3 element array of doubles); input-only

The Easting and Northing.

RetVal

Variant (3 element array of doubles)

The coordinates in XY.

Remarks

Xy must be declared as a variant and not as an array. To access the X and Y, subscript the variant (var(0), var(1)). The third element (elevation) is unchanged.

ElevationAt Method (Civil Engineering Feature)

Given a station, returns the elevation..

See Also | Example

Signature

RetVal = object.ElevationAt(Station)

object

EGProfile, FGProfile The object or objects this

property applies to.

Station

Double, input-only

The station to query for.

RetVal

Double

The elevation for the station.

ExternalStaToInternal Method

Given an external Station value (one using Station Equations), returns the internal Station value

See Also | Example

Signature

RetVal = object.ExternalStaToInternal(Station)

object

Alignment The object this property applies to.

Station

Double; read-only
The external station.

Variant (array of doubles)

RetVal

Variant

An array of internal station values.

Remarks

An External Station value represents one or more internal Station values. These values are returned in an array of doubles.

FindAllConnectingEdges Method

Returns all edges for a given Surface point.

See Also | Example

Signature

RetVal = object.FindAllConnectingEdges(Easting, Northing)

object

Surface The object or objects this property applies

to.

Easting

Double; input-only

The exact Easting of the Surface point.

Northing

Double; input-only

The exact Northing of the Surface point.

RetVal

Variant (array of doubles) An array of 3D points.

Remarks

FindAllConnectingEdges expects the Easting / Northing to be a Surface point. If not, the function will return with an error. The exact coordinates can be found using the Surface <u>FindPoint</u> method or by using a TinPoint

The first element in the array will be the coordinate passed in. The following items in the array are all of the point that radiate from the selected point. All edges are returned in counter-clockwise order.

FindAllFaces Method

Redefines the collection with all faces on the surface.

See Also | Example

Signature

object.FindAllFaces()

object

Faces The object or objects this property applies to.

Remarks

Use this method to reset the collection to all faces in the surface. The <u>SearchType</u> property will be set to kNoSearch since no filter is being applied.

FindConnectingEdge Method

Given an Surface point, retrieves the closest edge.

See Also | Example

Signature

RetVal = object.FindConnectingEdge(Easting, Northing)

object

Surface The object or objects this property applies

to.

Easting

Double; input-only

Easting of the Surface point.

Northing

Double; input-only

Northing of the Surface point.

RetVal

Variant (6 element array of doubles)

An array of two 3D points.

Remarks

FindConnectingEdge will return the closest edge to the input point.

FindFace Method

Given an Easting / Northing coordinate, redefines the collection with a single Face.

See Also | Example

Signature

object.FindFace(Easting, Northing)

object

Faces The object or objects this property applies to.

Easting

Double; input-only

The Easting coordinate to query.

Northing

Double; input-only

The Northing coordinate to query.

FindPath Method

Give two coordinates, reads in all of the Faces that lie on that line.

See Also | Example

Signature

object.FindPath(StartEasting, StartNorthing, EndEasting, EndNorthing)

object

Faces The object or objects this property applies to.

StartEasting

Double; input-only

The first point's Easting coordinate.

StartNorthing

Double; input-only

The first point's Northing coordinate.

EndEasting

Double; input-only

The second point's Easting coordinate.

EndNorthing

Double; input-only

The second point's Northing coordinate.

Remarks

FindPath will act as a filter for the Faces collection. Given two points, it

will return all of the faces that the line crosses. The Faces collection now contains only those faces. The <u>SearchType</u> will be set to kSearchByPath.

FindPoint Method

Given an Easting / Northing, returns the closest Surface point.

See Also | Example

Signature

RetVal = object.FindPoint(Easting, Northing)

object

Surface The object or objects this property applies

to.

Easting

Double; input-only

The Easting of the Surface point.

Northing

Double; input-only

The Northing of the Surface point.

RetVal

Variant (3 element array of doubles)

A 3D point representing the Surface point.

GetBoundingBox Method

Gets the bounding box for the surface.

See Also | Example

Signature

object.GetBoundingBox(LowerLeftPoint, UpperRightPoint)

object

Surface The object or objects this property applies

to.

LowerLeftPoint

Variant (3 element array of doubles); output-only The lower left coordinate for the bounding box in

Easting, Northing, Elevation format.

UpperRightPoint

Variant (3 element array of doubles); output-only The upper right coordinate for the bounding box in

Easting, Northing, Elevation format.

Remarks

All coordinates for the surface will lie within the bounding box. The surface must be open and built before using the GetBoundingBox method.

GetDouble Method

Gets the specified preference setting.

See Also | Example

Signature: Overview

1 Preferences Alignment

1 PreferencesCogo

1 <u>PreferencesCrossSection</u> (Civil Engineering Feature)

1 <u>PreferencesProfile</u> (Civil Engineering Feature)

1 PreferencesSurface

Gets the specified preference setting for Alignments.

Signature

RetVal = object.GetDouble(Setting)

object

<u>PreferencesAlignment</u> The object or objects this

property applies to.

Setting

eAeccPrefAlignDouble enum; input-only.

Specifies the setting to return.

kOffsetLeftDistOuter: Offset alignment - left

distance outer

kOffsetRightDistOuter: Offset alignment - right

distance outer

kOffsetLeftDistSecond: Offset alignment - left

distance second

kOffsetRightDistSecond:

Offset alignment - right

distance second

kOffsetLeftDistThird: Offset alignment - left

distance third

kOffsetRightDistThird:

Offset alignment - right

distance third

kOffsetLeftDistInner: Offset alignment - left

distance inner

kOffsetRightDistInner: Offset alignment - right

distance inner

kStationTickIncrement: Station labels tick

increment

kStationLabelIncrement: Station labels increment

kStationLabelOffset: Station labels offset

RetVal

Double

The value of the setting.

Gets the specified preference setting for Cogo.

Signature

RetVal = object.GetDouble(Setting)

object

PreferencesCogo

The object or objects this property applies to.

Setting

eAeccPrefCogoDouble enum; input-only.

Specifies the setting to return.

kPntCreateDefaultElev: Create default elevation

kPntInsertFixedElev: Insert fixed elevation

kPntInsertNoActualElev: Insert no elevation

kPntMarkerSize: Point marker size

kPntTextSize: Point text size

kPntTextRotation: Point text rotation

RetVal

Double

The value of the setting.

Gets the specified preference setting for Cogo.

Signature

RetVal = object.GetDouble(Setting)

object

PreferencesCrossSection

The object or objects this property applies to.

Setting

eAeccPrefCrossSectionDouble enum; input-only.

Specifies the setting to return.

kElevationInc: Elevation increment

kOffsetInc: Offset increment

RetVal

Double

The value of the setting.

Gets the specified preference setting for Vertical Alignments.

Signature

RetVal = object.GetDouble(Setting)

object

PreferencesProfile

The object or objects this property applies to.

Setting

eAeccPrefProfileDouble enum; input-only.

Specifies the setting to return.

kLabelInc: Label increment

kTangentLabelInc: Tangent label increment

kVerticalCurveLabelInc: Vertical curvel label

increment

kPassingSightDistEye: Passing sight eye height

distance

kPassingSightDistObj:

Passing sight object

height distance

kStoppingSightDistEye: Stopping sight eye height

distance

kStoppingSightDist_obj:

Stopping sight object

height distance

kKCrest: K minimum for crest

kKSag: K minimum for sag

kOverlap: Overlap

RetVal

Double

The value of the setting.

Gets the specified preference setting for Surfaces.

Signature

RetVal = object.GetDouble(Setting)

object

PreferencesSurface

The object or objects this property applies to.

Setting

eAeccPrefSurfaceDouble enum; input-only.

Specifies the setting to return.

Contour weeding kContWeedDist:

distance

kContWeedAngle: Contour weeding angle

Contour kContWeedSuppDist:

supplementing

distance

Contour kContWeedSuppBulge:

supplementing bulge

Surface vertical kVertExaggerationFactor:

exaggeration factor

Elevation begin range kElevRangeBegVal1:

range 1

Elevation begin range kElevRangeBegVal2:

range 2

Elevation begin range kElevRangeBegVal3:

range 3

Elevation begin range kElevRangeBegVal4:

range 4

Elevation begin range kElevRangeBegVal5:

range 5

Elevation begin range kElevRangeBegVal6:

range 6

Elevation begin range kElevRangeBegVal7:

range 7

kElevRangeBegVal8: Elevation begin range -

range 8

kElevRangeBegVal9: Elevation begin range -

range 9

kElevRangeBegVal10: Elevation begin range -

range 10

kElevRangeBegVal11: Elevation begin range -

range 11

kElevRangeBegVal12: Elevation begin range -

range 12

kElevRangeBegVal13: Elevation begin range -

range 13

kElevRangeBegVal14: Elevation begin range -

range 14

kElevRangeBegVal15: Elevation begin range -

range 15

kElevRangeBegVal16: Elevation begin range -

range 16

kElevRangeEndVal1: Elevation end range -

range 1

kElevRangeEndVal2: Elevation end range -

range 2

Elevation end range -

kElevRangeEndVal3: range 3

Elevation end range kElevRangeEndVal4:

range 4

Elevation end range kElevRangeEndVal5:

range 5

Elevation end range kElevRangeEndVal6:

range 6

Elevation end range kElevRangeEndVal7:

range 7

Elevation end range kElevRangeEndVal8:

range 8

Elevation end range kElevRangeEndVal9:

range 9

Elevation end range kElevRangeEndVal10:

range 10

Elevation end range kElevRangeEndVal11:

range 11

Elevation end range kElevRangeEndVal12:

range 12

Elevation end range kElevRangeEndVal13:

range 13

Elevation end range kElevRangeEndVal14:

range 14

Elevation end range kElevRangeEndVal15:

range 15

kElevRangeEndVal16: Elevation end range - range 16

kSurfDisplayBaseElev: Surface - display base

elevation

kContCreateMinElev: Contours create - minimum contour starting elevation

kContCreateMaxElev: Contours create - maximum contour starting elevation

kContCreateMinorInterval: Contours create - minor contour interval

kContCreateMinorWidth: Contours create - minor contour width

kContCreateMajorInterval: Contours create - major contour interval

kContCreateMajorWidth: Contours create - major contour width

kContLabelElevIncrement: Contours - label elevation increment

kContLabelSpacingDist: Contours - Label spacing distance

kWShedMinDepth: Water Sheds - minimum depth

Water Sheds -

kWShedMinArea: minimum area

RetVal

Double

The value of the setting.

GetElevation Method

Given a Northing and Easting, return the elevation.

See Also | Example

Signature

RetVal = object.GetElevation(Easting, Northing)

object

Surface The object or objects this property applies

to.

Easting

Double; input-only

The Easting coordinate to query.

Northing

Double; input-only

The Northing coordinate to query.

RetVal

Double

The elevation for the Easting / Northing.

Remarks

If the coordinates are within the bounds of the Surface, the elevation will be returned. If the coordinates lie outside of the Surface or lie in a hole in the Surface, then -1e20 will be returned.

GetInteger Method

Gets the specified preference setting.

See Also | Example

Signature: Overview

- 1 PreferencesAlignment
- 1 PreferencesCogo
- 1 PreferencesCrossSection (Civil Engineering Feature)
- 1 PreferencesParcel
- 1 <u>PreferencesProfile</u> (Civil Engineering Feature)
- 1 PreferencesSurface

Gets the specified preference setting for Alignments.

Signature

RetVal = object.GetInteger(Setting)

object

<u>PreferencesAlignment</u> The object or objects this property applies to.

Setting

eAeccPrefAlignInt enum; input-only. Specifies the setting to return.

kStationFmtUseStationFormat: Station format - use

Station format - use

negative kStationFmtUseNegParen:

parentheses

Station format kStationFmtDropDecimal:

drop decimal

Station format - use kStationFmtLeadZeros:

lead zeros

Station format kStationFmtCharPos:

character position

Station format kStationFmtFieldWidth:

field width

Station format kStationFmtDecimalPrec:

decimal precision

kSpiralType: Spiral type

kOffsetToggle: Offset toggle

RetVal

Integer

The value of the setting.

Gets the specified preference setting for Cogo.

Signature

RetVal = object.GetInteger(Setting)

object

<u>PreferencesCogo</u>

The object or objects this property applies to.

Setting

eAeccPrefCogoInt enum; input-only.

Specifies the setting to return.

kPntCreateNextNum: Next point number

kPntInsertAsCreated: Insert points into

drawing as created

kPntCreateNumSequential:

Point number

sequential

kPntCreateElevMode: Point - create

elevation mode

kPntCreateDescMode: Point - create

description mode

kPntInsertAtElevation: Point - insert at

elevation

kPntInsertAutoLabel: Point - insert

autolabel

kPntUpdateAllowMove:

Point update - allow

point move

Point update -

kPntUpdateDbAfterMove: update database

after move

kPntUpdateReuniteSym: Point update - reunite symbol

Point update - kPntUpdateCheckDbOnOpen: check database on

open

kPntCoordDispType: Point - coordinate

display type

kPntCoordEcho: Point - echo coordinates

kDscKeySearchOrder: Description Key - search order

kDscKeyExtendedSearch: Description Key - extended search

order

kDscKeyParamMatch: Description Key - Parameter match

kPntPrefCmdLineListOn:

Point - Command line list on

kPntPrefCmdLineGroupOn: Point - command line group on

kPntPrefAutoRegen: Point - auto regen

kPntPrefSortListRem: Point - sort list

kPntPrefSortListDuplicate: Point - sort list duplicate

kPntMarkerAcadPoint: Point marker - AutoCAD point

kPntMarkerBox: Point marker - box

kPntMarkerCircle: Point marker - circle

kPntMarkerStyle: Point marker - style

kPntMarkerAlignWithText:

Point marker - align

with text

kPntNumVisible: Point - number is

visible

kPntElevVisible: Point - elevation is

visible

kPntDescVisible: Point - description is

visible

kPntShowFullDesc: Point - show full

description

kPntNumColor: Point - number color

kPntElevColor: Point - elevation

color

kPntDescColor: Point - description

color

kPntAutoLeader: Point - auto leader

Integer

The value of the setting.

Gets the specified preference setting for Cross Sections.

Signature

RetVal = object.GetInteger(Setting)

object

PreferencesCrossSection

The object or objects this property applies to.

Setting

eAeccPrefCrossSectionInt enum; input-only.

Specifies the setting to return.

kDatumLayerOn: Datum layer visibility

kEGLabelPrecision Existing ground label

precision

kEGLayerOn: Existing ground layer

visibility

kElevationLabelInc: Elevation label

increment

kElevationPrecision: Elevation precision

kFGLabelPrecision Finished groung label

precision

kOffsetLabelInc: Offset label increment

kOffsetPrecision: Offset precision

kROWLinesLayerOn: Right of Way line layer

visibility

kRowsAboveMax: Rows above maximum

kRowsBelowDatum: Rows below datum

kSectionGridLayerOn: Section grid layer

visibility

kSectionGridTextLayerOn: Section grid text layer

visibility

kTemplateLayerOn: Template layer visibility

RetVal

Integer

The value of the setting.

Gets the specified preference setting for Parcels.

Signature

RetVal = object.GetInteger(Setting)

object

<u>PreferencesParcel</u>

The object or objects this property applies to.

Setting

eAeccPrefParcelsInt enum; input-only.

Specifies the setting to return.

kDefineAsSized: Define parcel as sized

kCheckAcrossChord: Map check across chord

kTruncateAreaLabels: Truncate area labels

kAutoLabelPlacement: Automatic label placement

kIncludeParcelLines: Include parcel lines on

import

kNumberLabelsOn: Parcel number label

visibility

kSequentialOn: Sequential parcel

numbering

kNextParcelNumber: Next parcel number

kSqUnitLabelsOn: Square unit label visibility

kSqUnitPrecision: Square unit precision

kAreaUnitLabelsOn: Area unit label visibility

kAreaUnitPrecision: Area unit precision

RetVal

Integer

The value of the setting.

Gets the specified preference setting for Vertical Alignments.

Signature

RetVal = object.GetInteger(Setting)

object

PreferencesProfile

The object or objects this property applies to.

Setting

eAeccPrefProfileInt enum; input-only.

Specifies the setting to return.

kEgPrecision: Existing ground label precision

kFgPrecision: Finished ground label precision

kGridDsp: 0: Grid creation is off

1: Grid creation is on

klr_on:

0: Left to right creation is set

1: Right to left creation is set

RetVal

Integer

The value of the setting.

Gets the specified preference setting for Surfaces.

Signature

RetVal = object.GetInteger(Setting)

object

PreferencesSurface

The object or objects this property applies to.

Setting

eAeccPrefSurfaceInt enum; input-only.

Specifies the setting to return.

kviewtype: View type

kRangeNumber: Range number

kfltdel: Fault delete

krngs_type: Ranges type

kRange1Color: Range color 1

kRange2Color: Range color 2

kRange3Color: Range color 3

kRange4Color: Range color 4

kRange5Color: Range color 5

kRange6Color: Range color 6

kRange7Color: Range color 7

kRange8Color: Range color 8

kRange9Color: Range color 9

kRange10Color: Range color 10

kRange11Color: Range color 11

kRange12Color: Range color 12

kRange13Color: Range color 13

kRange14Color: Range color 14

Range color 15 kRange15Color:

kRange16Color: Range color 16

kGraphicsColor: **Graphics color**

Contour kContSmoothFactor:

smoothing factor

Contour kContSmoothType:

smoothing type

Contour - label kContLabelPrecision:

precision

Contour - create as kContCreateAsPoly:

polyline

Contour - label kContLabelMultipleToggle:

multiple

Contour - style kContStyleMgrLastUsedPage:

manager last used

page

Watershed - exceed kWShedExceedBothCriteria:

both criteria

Watershed - erase kWShedErasePrevious:

previous

Watershed - display kWShedDisplayNumber:

number

Watershed - use

kWShedSolidFill: solid fill

RetVal

Integer
The value of the setting.

GetLayerName Method (Civil Engineering Feature)

Returns the layer names of the alignment cross section or alignment profile.

See Also | Example

Signature: Overview

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 ProfileBlock (Civil Engineering Feature)

Returns the layer names of the alignment cross section.

Signature

RetVal = object.GetLayerName(val)

object

CrossSectionBlock The object or objects this

property applies to.

val

eAeccCrossSectionBlockLayer enum; input-only.

Specifies the layer name to return.

kCrossSectionEGTextLayer: Existing grade text

layer

kCrossSectionEGLayer: Existing grade layer

kCrossSectionFGLayer: Finished grade layer

kCrossSectionFGTextLayer: Finished grade text

layer

kCrossSectionGridLayer: Grid layer

kCrossSectionGridTextLayer: Grid text layer

RetVal

String

Returns the Layer Name of the CrossSectionBlock.

Returns the layer names of the alignment profile.

Signature

RetVal = object.GetLayerName(val)

object

ProfileBlock

The object or objects this property applies to.

val

eAeccProfileBlockLayer enum; input-only.

Specifies the layer name to return.

kBaseVertical: Existing ground base vertical

layer

kBaseHorizontal: Existing ground base horizontal

layer

kDitchLeft: Finished ground ditch left layer

kDitchRight: Finished ground ditch right layer

Existing groung center surface kEGC:

layer

Existing ground center surface kEGCText:

text layer

Existing ground left surface kEGL:

layer

Existing ground left surface text kEGLText:

layer

Existing ground right surface kEGR:

layer

Existing ground right surface kEGRText:

text layer

Finished ground center kFGC:

Finished ground center text kFGCText:

Finished ground left trans 1 kFGL1:

layer

Finished ground left trans 2 kFGL2:

layer

Finished ground left trans 3 kFGL3:

layer

Finished ground left trans 4 kFGL4:

layer

kFGL5: Finished ground left trans 5

layer

kFGL6: Finished ground left trans 6

layer

kFGL7: Finished ground left trans 7

layer

kFGL8: Finished ground left trans 8

layer

kFGR1: Finished ground right trans 1

layer

kFGR2: Finished ground right trans 2

layer

kFGR3: Finished ground right trans 3

layer

kFGR4: Finished ground right trans 4

layer

kFGR5: Finished ground right trans 5

layer

kFGR6: Finished ground right trans 6

layer

kFGR7: Finished ground right trans 7

layer

kFGR8: Finished ground right trans 8

layer

kGrid: Existing ground grid layer

kGridText: Existing ground text layer

kStationText: Existing ground station text

RetVal

String

Returns the Layer Name of the ProfileBlock.

GetStaStrWithEquations Method

Given an internal station value, returns that station as a formatted string with Station Equations applied.

See Also | Example

Signature

RetVal = object.GetStaStrWithEquations(Station)

object

Alignment The object or objects this property

applies to.

Station

Double; input-only Internal station value.

RetVal

String

The formatted station string.

GetStaWithEquations Method

Given an internal station value, returns that station value with Station Equations applied.

See Also | Example

Signature

RetVal = object.GetStaWithEquations(Station)

object

Alignment The object or objects this property

applies to.

Station

Double; input-only Internal station value.

RetVal

Double

The station number.

GetString Method

Gets the specified preference setting.

See Also | Example

Signature: Overview

- 1 PreferencesAlignment
- 1 PreferencesCogo
- 1 <u>PreferencesCrossSection</u> (Civil Engineering Feature)
- 1 PreferencesParcel
- 1 <u>PreferencesProfile</u> (Civil Engineering Feature)
- 1 PreferencesSurface

Gets the specified preference setting for Alignments.

Signature

RetVal = object.GetString(Setting)

object

<u>PreferencesAlignment</u> The object or objects this property applies to.

Setting

eAeccPrefAlignString enum; input-only. Specifies the setting to return.

kStationFmtDecChar: Station format -

decimal character

kStationFmtStaChar: Station format - station

character

kAlignLayerPrefix: Alignment layer prefix

kLabelStaEquAhead: Label station equation

ahead

kLabelStaEquBack: Label station equation

back

kLabelPT: Label PT

kLabelPC: Label PC

kLabelCPI: Label CPI

kLabelPCC: Label PCC

kLabelPRC: Label PRC

kLabelTS: Label TS

kLabelSC: Label SC

kLabelCS: Label CS

kLabelST: Label ST

kLabelSPI: Label SPI

kLabelCC: Label CC

kLabelPI: Label PI

kLabelSS: Label SS

kOffsetLeftLayerOuter: Offsets layer - left outer

kOffsetLeftLayerSecond:
Offsets layer - left

second

kOffsetLeftLayerThird: Offsets layer - left third

kOffsetLeftLayerInner: Offsets layer - left inner

kOffsetLeftNameOuter: Offsets name - left

outer

kOffsetRightNameOuter: Offsets name - right

name

kOffsetLeftNameSecond: Offsets name - left

second

kOffsetRightNameSecond: Offsets name - right

second

kOffsetLeftNameThird: Offsets name - left third

kOffsetRightNameThird:

Offsets name - right

third

kOffsetLeftNameInner: Offsets name - left

inner

kOffsetRightNameInner: Offsets name - right

inner

kOffsetNamePrefix: Offsets name - prefix

kStationLayer: Station layer

kStationPntLayer: Station point layer

kStationEquLayer: Station equations layer

RetVal

String

The value of the setting.

Gets the specified preference setting for Cogo.

Signature

RetVal = object.GetString(Setting)

object

PreferencesCogo

The object or objects this property applies to.

Setting

eAeccPrefCogoString enum; input-only.

Specifies the setting to return.

kPntCreateDefaultDesc: Point - Create default

description

kDscKeySymPath: Description Key symbol

path

kPntTextStyle: Point - text style

String

The value of the setting.

Gets the specified preference setting for Cross Sections.

Signature

RetVal = object.GetString(Setting)

object

PreferencesCrossSection

The object or objects this property applies to.

Setting

eAeccPrefCrossSectionString enum; input-only.

Specifies the setting to return.

kDatumLayer: Datum layer

kEGLayer: Existing grade layer

kROWLinesLayer: Right of Way lines layer

kSectionGridLayer: Section grid layer

kSectionGridTextLayer: Section grid text layer

kTemplateLayer: Template layer

kTemplatePath: Cross section template

path

RetVal

String

The value of the setting.

Gets the specified preference setting for Parcels.

Signature

RetVal = object.GetString(Setting)

object

PreferencesParcel

The object or objects this property applies to.

Setting

eAeccPrefParcelsString enum; input-only.

Specifies the setting to return.

kNumberPrefix: Parcel numbering prefix

kNumberTextStyle: Parcel numbering text style

kSqUnitTextStyle: Square unit text style

kSqUnitSuffix: Square unit suffix

kAreaUnitTextStyle: Area unit text style

kAreaUnitSuffix: Area unit suffix

kParcelLayer: Parcel layer

kLabelLayer: Parcel label layer

RetVal

String

The value of the setting.

Gets the specified preference setting for Vertical Alignments.

Signature

RetVal = object.GetString(Setting)

object

PreferencesProfile

The object or objects this property applies to.

Setting

eAeccPrefProfilelString enum; input-only.

Specifies the setting to return.

kTextLayer: Existing ground station text

layer

kVertGridLayer: Existing ground vertical grid

layer

kBaseGridLayer: Existing ground base grid

layer

kEgLayer: Existing groung center layer

kEgLeftLayer: Existing ground left layer

kEgRightLayer: Existing ground right layer

kFgLayer: Finished ground center layer

kGridLayer: Existing ground grid layer

kGridTextLayer: Existing ground grid text layer

Existing groung center text kEgtLayer:

layer

kEglefttLayer: Existing ground left text layer

Existing ground right text layer kEgRighttLayer:

kDitchLeftLayer: Finished ground ditch left layer

Finished ground ditch right kDitchRightLayer:

layer

Finished ground text layer kFgtLayer:

Beginning vertical curve kBvcs:

station label text

Beginning vertical curve kBvce:

elevation text

End vertical curve station label kEvcs:

text

End vertical curve elevation kEvce:

text

Sample profile direction kProfileDir:

Finished ground left 1 trans kFgLeftLayer1:

layer

Finished ground left 2 trans kFgLeftLayer2:

layer

Finished ground left 3 trans kFgLeftLayer3:

layer

kFgLeftLayer4: Finished ground left 4 trans

layer

kFgLeftLayer5: Finished ground left 5 trans

layer

kFgLeftLayer6: Finished ground left 6 trans

layer

kFgLeftLayer7: Finished ground left 7 trans

layer

kFgLeftLayer8: Finished ground left 8 trans

layer

kFgRightLayer1: Finished ground right 1 trans

layer

kFgRightLayer2: Finished ground right 2 trans

layer

kFgRightLayer3: Finished ground right 3 trans

layer

kFgRightLayer4: Finished ground right 4 trans

layer

kFgRightLayer5: Finished ground right 5 trans

layer

kFgRightLayer6: Finished ground right 6 trans

layer

kFgRightLayer7: Finished ground right 7 trans

layer

kFgRightLayer8: Finished ground right 8 trans

layer

kPrefix: Profile layer prefix

kHighPoint: High point label text

kLowPoint: Low point label text

kPviLabel: Point of vertical intersection

(PVI) text

kADiff: Algebraic difference (A.D.) text

kKValue: Curve coefficient (K) text

RetVal

String

The value of the setting.

Gets the specified preference setting for Surfaces.

Signature

RetVal = object.GetString(Setting)

object

PreferencesSurface

The object or objects this property applies to.

Setting

eAeccPrefSurfaceString enum; input-only.

Specifies the setting to return.

kSlopeRangeBegVal1: Slope range - begin 1

kSlopeRangeBegVal2: Slope range - begin 2

kSlopeRangeBegVal3: Slope range - begin 3

kSlopeRangeBegVal4: Slope range - begin 4

kSlopeRangeBegVal5: Slope range - begin 5

kSlopeRangeBegVal6: Slope range - begin 6

kSlopeRangeBegVal7: Slope range - begin 7

kSlopeRangeBegVal8: Slope range - begin 8

kSlopeRangeBegVal9: Slope range - begin 9

kSlopeRangeBegVal10: Slope range - begin 10

kSlopeRangeBegVal11: Slope range - begin 11

kSlopeRangeBegVal12: Slope range - begin 12

kSlopeRangeBegVal13: Slope range - begin 13

kSlopeRangeBegVal14: Slope range - begin 14

kSlopeRangeBegVal15: Slope range - begin 15

kSlopeRangeBegVal16: Slope range - begin 16

kSurfaceLayer: Surface layer

kBoundaryLayer: Boundary layer

kRange1Layer: Range layer 1

kRange2Layer: Range layer 2

kRange3Layer: Range layer 3

kRange4Layer: Range layer 4

kRange5Layer: Range layer 5

kRange6Layer: Range layer 6

kRange7Layer: Range layer 7

kRange8Layer: Range layer 8

kRange9Layer: Range layer 9

kRange10Layer: Range layer 10

kRange11Layer: Range layer 11

kRange12Layer: Range layer 12

kRange13Layer: Range layer 13

kRange14Layer: Range layer 14

kRange15Layer: Range layer 15

kRange16Layer: Range layer 16

kBreakLineLayer: BreakLine layer

kLayerPrefix: Surface Layer

prefix

kContCreateMinorLayer: Contour - create

minor layer

kContCreateMajorLayer: Contour - create

major layer

kContourStyle: Contour - style

kWShedLayerBoundPnt: Watershed layer -

bounding points

bounding segments

kWShedLayerNonBound: Watershed layer -

nonbound

kWShedLayerDepression:Watershed layer -

depression

Watershed layer kWShedLayerFlatArea:

flat area

Watershed layer kWShedLayerMultiDrain:

multidrain

Watershed layer kWShedLayerLocalMin: local minimum

Watershed layer kWShedLayerDrain:

drain

Watershed layer kWShedLayerMultiDrainNotch:

multidrain notch

RetVal

String

The value of the setting.

Import Method

Draws the alignment, cross section, existing ground profile, parcel, or surface into the current drawing.

See Also | Example

Signature: Overview

1 Alignment

1 <u>CrossSection</u> (Civil Engineering Feature)

l <u>EGProfile</u> (Civil Engineering Feature)

1 Parcel

1 Surface

Draws an alignment into the current drawing.

Signature

object.Import()

object

Alignment The object or objects this property applies to.

Remarks

Import will draw the alignment in the current drawing using the current

layer. When an alignment is imported, external data is stored on the AutoCAD entities and the entity ids are stored externally. If you want to be able to select an alignment from a drawing, you need to use the Import command. If you draw the alignment using your own routines, this extra information will not be supplied.

Draws an alignment cross section into the current drawing.

Signature

object.Import(Coordinates[, bBlockOnly])

object

CrossSection

The object or objects this property applies to.

Coordinates

Variant (array of doubles); input-only The starting point in X, Y format.

bBlockOnly

Boolean; input only; optional True: Only the block is inserted...

False: The cross section is drawn along with the

block (default).

Draws the existing ground profile as a Quick Profile in the current drawing.

Signature

object.Import(Coordinates[, StartingStation][, EndingStation][, DatumElevation][, VerticalScale][, bLeftToRight][, bBlockOnly)

object

EGProfile

The object or objects this property applies to.

Coordinates

Variant (2 element array of doubles); input-only

The starting point in X, Y format.

StartingStation

Double; input only; optional

The starting station of the alignment. The default is

the StartingStation of the Alignment object.

EndingStation

Double; input only; optional

The ending station of the alignment. The default is

the EndingStation of the Alignment object.

DatumElevation

Double; input only; optional

The elevation of the base line for the profile. The default value is calculated using the minimum

elevation of the profile.

VerticalScale

Double; input only; optional

The vertical scale applied to the profile. The default

value is the VerticalScale of the DatabasePreferences object.

bLeft To Right

Boolean; input only; optional

True: The profile is drawn left to right (default).

False: The profile is drawn right to left.

bBlockOnly

Boolean; input only; optional

True: Only the block is inserted..

False: The profile is drawn along with the block

(default).

Remarks

You may import a profile for center, left or right existing ground. If you import more than 1 profile for a given alignment, you can identify it using the Coordinates property of the ProfileBlock object (note that the Land Desktop commands do not support more than one profile per alignment).

Draws a parcel into the current drawing.

Signature

object.Import()

object

Parcel

The object or objects this property applies to.

Draws a surface in the current drawing as 3D surfaces.

Signature

object.Import()

object

Surface

The object or objects this property applies to.

Remarks

The Import function is not supported for Volume Surfaces (Composite/Diff Grid).

InstantGrade Method

Given a station, returns the instantaneous grade and the algebraic difference.

See Also | Example

Signature: Overview

l <u>EGProfile</u> (Civil Engineering Feature)

1 <u>FGProfile</u> (Civil Engineering Feature)

Signature

object.InstantGrade(Station, Grade)

object

EGProfile The object or objects this property applies

to.

Station

Double; input-only

The station to query for.

Grade

Double; output-only

The grade at the station.

Signature

object.InstantGrade(Station, Grade, Diff)

object

FGProfile

The object or objects this property applies to.

Station

Double; input-only

The station to query for.

Grade

Double; output-only

The grade at the station.

Diff

Double; output-only

The algebraic difference between the two grades.

Remarks

The algebraic difference will only exist for a vertical curve.

Item Method

Gets the member object at a given index in a collection.

See Also | Example

Signature

RetVal = object.Item(Index)

object

All Collections The object or objects this property

applies to.

Index

Variant; input-only

The index location in the collection for the member

item to query.

The index must be either an integer or a string. If an

integer, the index must be between 0 and N-1,

where N is the number of objects in the collection or

selection set.

RetVal

Object;

The object at the given index location in the

collection or selection set.

Remarks

This method supports string based iteration. For example, you could reference the document named "siteplan.dwg" through the following

statement:

```
Set mydoc = Docs.Item("siteplan.dwg")
```

In general, the index can be passed in as any integral type: integer, long, double, etc. In addition, the following collections support a case-sensitive string type as an index (typically the name of the collection member):

Boundaries

BreakLines

ContourItems

DEMFiles

DescriptionKeyFile

DescriptionKeyFiles

Drawings

PointGroupNames

PointFiles

Projects

Prototypes

WaterSheds

Also, the following collections support a case-insensitive string type as an index (typically the name of the collection member):

Alignments

Parcels

<u>PointGroups</u>

Surfaces

If a station equation is deleted, then the indices for the <u>StationEquations</u> collection will change.

LineIntersection Method

Returns the intersection points of a line with the Alignment.

See Also | Example

Signature

RetVal = object.LineIntersection(StartEasting, StartNorthing, EndEasting, EndNorthing)

object

Alignment The object or objects this property

applies to.

StartEasting

Double; input-only

Easting of the first point

StartNorthing

Double; input-only

Northing of the first point

EndEasting

Double; input-only

Easting of the second point

EndNorthing

Double; input-only

Northing of the second point

RetVal

Variant (array of doubles) The array has the format of: Station, Direction, Easting, Northing.

LoadSetupProfile Method

Load the specified drawing setup profile.

See Also | Example

Signature

object.LoadSetupProfile(ProfileName)

object

DatabasePreferences The object or objects this

property applies to.

ProfileName

String; input-only

The profile name to load.

Remarks

The drawing setup files can be found at the path specified by the DrawingSetupPath property of the PreferencesFiles object.

LockPoints Method

Locks a list of point numbers.

See Also | Example

Signature

object.LockPoints(PointString)

object

CogoPoints The object or objects this property

applies to.

PointString

String; input-only

Comma delimited string of points to be locked with

groups separated by hyphens.

Remarks

Typical examples of the point string format are: "1,2,3,5,10" "1-100,1000-2000"

NewProjectBased Method

Creates a new AutoCAD Land Desktop document.

See Also | Example

Signature

RetVal = object.NewProjectBased(TemplateName, DrawingName[, ProjectName])

object

Document The object or objects this property

applies to.

TemplateName

String; input-only

The file name of the template.

DrawingName

String; input-only

The file name for the new drawing.

ProjectName

String; input-only; optional The name of the Project

RetVal

Document

The newly created Document object.

Remarks

The drawing name can be up to 255 characters, including path and file extension, and it must be unique. You cannot create two drawings with the same name in the same drawing folder.

It is not necessary to add .dwg to the end of the name; the drawing file extension is created automatically. If a path is not specified, the new drawing will be created at the current <u>DrawingPath</u> for the <u>ActiveProject</u>.

It is not necessary to add .dwt to the end of the TemplateFileName; it will be added automatically if not specified. If the path is not specified, the TemplateDwgPath of the AcadPreferences object will be used.

If a path for the Project is not specified, the <u>Path</u> of the <u>ActiveProject</u> object is used.

OffsetElevationToXy Method (Civil Engineering Feature)

Given an offset and elevation, return the AutoCAD XY coordinates.

See Also | Example

Signature

RetVal = object.OffsetElevationToXy(OffsetElevation)

object

CrossSectionBlock The object or objects this

property applies to.

OffsetElevation

Variant (2 element array of doubles); input-only

The offset and elevation to query for.

Retval

Variant (2 element array of doubles); output-only

The XY coordinates for the given offset and

elevation.

Open Method

Opens an existing AutoCAD Land Desktop drawing.

See Also | Example

Signature

object.Open(DrawingName)

object

Drawings The object or objects this property applies

to.

DrawingName

String; input-only

The name of the drawing to open.

Remarks

It is recommended that you save the current drawing before calling this method.

If the drawing is associated with a project name that is different than the <u>Project</u>, then you will be prompted to choose a project.

It is not necessary to add .dwg to the end of the name; the drawing file extension will be added automatically if not specified. It is not necessary to specify a path; the current Path for the Drawings collection will be used if a path is not specified.

OpenProjectBased Method

Opens an existing AutoCAD Land Desktop drawing file (DWG).

See Also | Example

Signature

RetVal = object.OpenProjectBased(DrawingName[, ProjectName])

object

Document The object or objects this property

applies to.

DrawingName

String; input-only

The file name for the drawing.

ProjectName

String; input-only; optional The name of the project

RetVal

Document

The newly opened **Document** object.

Remarks

It is recommended that you save the current drawing before calling this method.

It is not necessary to add .dwg to the end of the name; the drawing file

extension will be added automatically if not specified. If a path is not specified, the path used will be the current <u>DrawingPath</u> for the <u>ActiveProject</u>.

If a path for the Project is not specified, the <u>Path</u> of the <u>ActiveProject</u> object is used.

Paste Method

The Paste method will take another Surface an paste it onto the given Surface.

See Also | Example

Signature

object.Paste(ChildSurface)

object

Surface The object or objects this property applies

to.

ChildSurface

String; input-only

Name of the Surface to be added to the given

Surface.

PerpIntersection Method

Returns the perpendicular intersection of a point with an Alignment.

See Also | Example

Signature

RetVal = object.PerpIntersection(Easting, Northing)

object

Alignment The object or objects this property

applies to.

Easting

Double; input-only Easting of the point

Northing

Double; input-only Northing of the point

RetVal

Variant (array of doubles)

The array has the format of: Station, Direction,

Easting, Northing.

PointByNumber Method

Returns a point in the project database by number.

See Also | Example

Signature

RetVal = object.PointByNumber(Number)

object

CogoPoints The object or objects this property

applies to.

Number

Long; input-only

The point number to query for.

RetVal

AeccPoint object The point object.

Remarks

If the point does not exist, then no object will be returned.

PointCodeDescription Method (Civil Engineering Feature)

Returns the description of a cross section point code.

See Also | Example

Signature

RetVal = object.PointCodeDescription(Code)

object

CrossSections The object or objects this property

applies to.

Code

Integer; input-only

The code number to query for.

RetVal

String;

The description of the point code.

PointLocation Method

Given a station and offset, returns the Northings and Eastings.

See Also | Example

Signature

object.PointLocation(Station, Offset, Easting, Northing, Direction)

object

Alignment The object or objects this property

applies to.

Station

Double; input-only

The Station to query for.

Offset

Double; input-only

The Offset to query for.

Easting

Double; output-only

The Easting of the coordinate.

Northing

Double; output-only

The Northing of the coordinate.

Direction

Double; output-only

The direction of the queried point.

Remarks

If the Offset is to the left of the alignment, it should be negative. If the Offset is to the right of the alignment, it should be positive.

PointNumberFromObjID Method

Given a AutoCAD object ID, PointNumberFromObjID will return the point number.

See Also | Example

Signature

RetVal = object.PointNumberFromObjID(ObjectID)

object

CogoPoints The object or objects this property

applies to.

ObjectID

Long; input-only

The AutoCAD entity's object ID. The object is must

be a valid AECC POINT.

RetVal

Long

The point number.

Remarks

The object ID must be for a valid AECC_POINT. If the object is not an AECC_POINT, then the function will return 0 which is an invalid point number.

PointStringToArray Method

Given a point string, returns an array with all of the point numbers.

See Also | Example

Signature

RetVal = object.PointStringToArray(PointString)

object

CogoPoints The object or objects this property

applies to.

PointString

String; input-only

Comma delimited string of points to be locked with

groups separated by hyphens.

RetVal

Variant (array of longs)

Each long is a point number.

Remarks

Typical examples of the point string format are: "1,2,3,5,10" "1-100,1000-2000"

ProfileByType Method

Returns the Existing or Finished Ground Profile by type.

See Also | Example

Signature: Overview

1 <u>EGProfiles</u> (Civil Engineering Feature)

1 FGProfiles (Civil Engineering Feature)

Returns the Existing Ground Profile by type.

Signature

RetVal = object.ProfileByType(Type, SurfaceName)

Object

EGProfiles The object or objects this method

applies to.

Type

eAeccEGProfileType enum; input-only

Returns the Existing Ground Profile by type.

kEgCenter: Existing ground center profile type

kEgLeft: Existing ground left profile type

kEgNone: Existing ground none profile type

. . .

kEgRight: Existing ground right profile type

SurfaceName

String; read-only

The name of the surface.

RetVal

EGProfile object;

The newly added **EGProfile** object.

Returns the Finished Ground Profile by type.

Signature

RetVal = object.ProfileByType(Type)

Object

FGProfiles

The object or objects this method applies to.

Type

eAeccFGProfileType enum; input-only

Returns the Finished Ground Profile by type.

kFgCenter: Finished ground profile center

type

kFgDitchLeft: Finished ground ditch left type

kFgDitchRight: Finished ground ditch right type

kFgLeft1: Finished ground left 1 trans type

kFgLeft2: Finished ground left 2 trans type

kFgLeft3: Finished ground left 3 trans type

kFgLeft4: Finished ground left 4 trans type

kFgLeft5: Finished ground left 5 trans type

kFgLeft6: Finished ground left 6 trans type

kFgLeft7: Finished ground left 7 trans type

kFgLeft8: Finished ground left 8 trans type

kFgNone: Finished ground none type

kFgRight1: Finished ground right 1 trans type

kFgRight2: Finished ground right 2 trans type

kFgRight3: Finished ground right 3 trans type

kFgRight4: Finished ground right 4 trans type

kFgRight5: Finished ground right 5 trans type

kFgRight6: Finished ground right 6 trans type

kFgRight7: Finished ground right 7 trans type

kFgRight8: Finished ground right 8 trans type

RetVal

FGProfile object;

The newly added <u>FGProfile</u> object.

RemoveAll Method

Removes all method for Alignments and StationEquations.

See Also | Example

Signature: Overview

1 Alignment

1 StationEquations

Removes all of the entities from the alignment.

Signature

object.RemoveAll()

object

Alignment The object or objects this property applies to.

Remarks

Before changing / adding any entities for an alignment, all entities must be removed.

Removes all station equations from the alignment.

Signature

object.RemoveAll()

object

<u>StationEquations</u>
The object or objects this property applies to.

RemoveAllLabels Method

Removes all contour labels.

See Also | Example

Signature

object.RemoveAllLabels()

object

AeccContour The object or objects this property applies to.

RemoveLabelAt Method

Removes the contour label closest to the specified location.

See Also | Example

Signature

object.RemoveLabelAt(X, Y)

object

AeccContour The object or objects this property

applies to.

X

Double; input-only

The X coordinate of the label.

Y

Double; input-only

The Y coordinate of the label.

Remarks

The label closest to the X, Y point will be removed from the Contour object.

Rename Method

Renames a surface or parcel.

See Also | Example

Signature: Overview

1 Parcels

1 Surfaces

Renames a parcel.

Signature

object.Rename(OldName, NewName)

object

Parcels The object or objects this property applies

to.

OldName

String; input-only

The name of the surface to be renamed.

NewName

String; input-only

The new name of the parcel.

Renames a surface.

Signature

object.Rename(OldName, NewName)

object

Surfaces

The object or objects this property applies to.

OldName

String; input-only

The name of the surface to be renamed.

NewName

String; input-only

The new name of the surface.

SampleElevations Method

Given a start and end point, returns an array of derived points from the Surface.

See Also | Example

Signature

RetVal = object.SampleElevations(StartPoint, EndPoint)

object

Surface The object or objects this property applies

to.

StartPoint

Variant (3 element array of doubles); input-only

The starting point to be queried in Easting, Northing,

Elevation.

EndPoint

Variant (3 element array of doubles); input-only

The ending point to be queried in Easting, Northing,

Elevation.

RetVal

Variant (array of doubles)

The sample points returned as Easting, Northing,

Elevation, Easting, etc.

Remarks

Given a starting and ending point, SampleElevations will return all of the points between the start and end. These points occur when a Surface Edge is crossed. The net result is that the function returns a line that is projected onto the Surface.

SampleElevations will return a Variant. This variant will be an array of doubles. Every 3 doubles is a point of the format Easting, Northing, Elevation.

Save Method

Writes objects to specific databases.

See Also | Example

Signature: Overview

l Alignment

l CogoPoint

l DescriptionKey

l PointGroup

Saves the Alignment.

Signature

object.Save()

object

Alignment The object or objects this property applies to.

Writes the CogoPoint to the project database.

Signature

object.Save()

object

CogoPoint

The object or objects this property applies to.

Writes the DescriptionKey to the project database.

Signature

object.Save()

object

DescriptionKey

The object or objects this property applies to.

Writes the PointGroup to the project database.

Signature

object.Save()

object

PointGroup

The object or objects this property applies to.

SaveAsDefault Method

Saves the DatabasePreferences to the registry.

See Also | Example

Signature

object.SaveAsDefault()

object

<u>DatabasePreferences</u> The object or objects this property applies to.

SaveSetupProfile Method

Saves the current drawing settings to a drawing setup profile.

See Also | Example

Signature

object.SaveSetupProfile(ProfileName)

object

DatabasePreferences The object or objects this

property applies to.

ProfileName

String; input-only

The profile name to save to.

Remarks

After you change any of the properties, use the SaveSetupProfile method to save these settings in a drawing setup profile. The new drawing setup profile will be saved at the path specified by the DrawingSetupPath property of the PreferencesFiles object.

SectionByStation Method (Civil Engineering Feature)

Returns a Cross Section object by specifing its station.

See Also | Example

Signature

RetVal = object.SectionByStation(Station)

object

CrossSections The object or objects this property

applies to.

Station

Double; input-only

The station for the desired cross section.

RetVal

CrossSection object

The Cross Section object referenced by the station.

SectionVolume Method (Civil Engineering Feature)

Calculates the cross section volumes.

See Also | Example

Signature

object.SectionVolume(Type, CurveCorrection, Swell, Shrink, CutArea, CutCenter, CutVolume, FillArea, FillCentroid, FillVolume)

object

CrossSection The object or objects this property

applies to.

Type

eAeccCrossSectionVolumeMethod enum; input-

only.

Specifies the setting to return.

kVolumeByAvgEndArea: Volume by average end

area calculation

kVolumeByPrismodal Volume by prismodal

calculation

CurveCorrection

Double; input-only

Applies a correction factor to the volume equal to the ratio of the centroid to the radius of the curve.

1 - curve correction on

0 - curve correction off

Swell

Double; input-only The swell factor

Shrink

Double; input-only The shrink factor

CutArea

Double; output-only

The cut area

CutCentroid

Double; output-only The cut centroid

CutVolume

Double; output-only

The cut volume

FillArea

Double; output-only

The fill area

FillCentroid

Double; output-only

The fill centroid

FillVolume

Double; output-only

The fill volume

SetBoundingBox Method

Sets the bounding box for the surface.

See Also | Example

Signature

object.SetBoundingBox(LowerLeftPoint, UpperRightPoint)

object

Surface The object or objects this property applies

to.

LowerLeftPoint

Variant (3 element array of doubles); input-only The lower left coordinate for the bounding box in

Easting, Northing, Elevation format.

UpperRightPoint

Variant (3 element array of doubles); input-only The upper right coordinate for the bounding box in

Easting, Northing, Elevation format.

Remarks

Calculations will be much faster if a bounding box is set. If no bounding box is set for a surface, then one will be generated when the surface is built.

SetDouble Method

Sets the specified preference setting.

See Also | Example

Signature

object.SetDouble(Setting, Value)

object

PreferencesAlignment,

PreferencesCrossSection, (Civil Engineering

Feature)

PreferencesCogo,

PreferencesProfile, (Civil Engineering Feature)

<u>PreferencesSurface</u>

The object or objects this property applies to.

Setting

eAeccPrefAlignDouble, eAeccPrefCogoDouble,

eAeccPrefCrossSectionDouble,

eAeccPrefProfileDouble, eAeccPrefSurfaceDouble

enum; input-only

Specifies the setting to set.

Value

Double; input-only

The value of the setting.

Remarks

For a listing of settings see the $\underline{\text{GetDouble}}$ method.

SetInteger Method

Sets the specified preference setting.

See Also | Example

Signature

object.SetInteger(Setting, Value)

object

PreferencesAlignment,

PreferencesCogo,

PreferencesCrossSection, (Civil Engineering

Feature)

PreferencesParcel,

PreferencesProfile, (Civil Engineering Feature)

PreferencesSurface,

The object or objects this property applies to.

Setting

eAeccPrefAlignInt, eAeccPrefCogoInt,

eAeccPrefCrossSectionInt,

eAeccPrefParceIInt eAeccPrefProfileInt, eAeccPrefSurfaceInt,

enum; input-only

Specifies the setting to set.

Value

Integer; input-only

The value of the setting.

Remarks

For a listing of settings see the <u>GetInteger</u> method.

SetReferenceCurve Method

Sets the reference curve for the Curve Text.

See Also | Example

Signature

object. Set Reference Curve (Circle Or Arc Entity)

object

AeccCurveText The object or objects this property

applies to.

CircleOrArcEntity

AcadEntity object; input-only

A circle or arc entity.

SetString Method

Sets the specified preference setting.

See Also | Example

Signature

object.SetString(Setting, Value)

object

<u>PreferencesAlignment</u>, <u>PreferencesCogo</u>,

PreferencesCrossSection, (Civil Engineering

Features)

PreferencesParcel,

PreferencesProfile, (Civil Engineering Features)

PreferencesSurface,

The object or objects this property applies to.

Setting

eAeccPrefAlignString, eAeccPrefCogoString,

eAeccPrefCrossSectionString,

eAeccPrefParcelString eAeccPrefProfileString, eAeccPrefSurfaceString,

enum; input-only

Specifies the setting to set.

Value

String; input-only

The value of the setting.

Remarks

For a listing of settings see the <u>GetString</u> method.

StationElevationToXy Method (Civil Engineering Features)

Given a station and elevation, return the AutoCAD XY coordinates.

See Also | Example

Signature

RetVal = object.StationElevationToXy(StaElev)

object

<u>ProfileBlock</u> The object or objects this property

applies to.

StaElev

Variant (2 element array doubles); input-only

The Station and Elevation to query for.

Retval

Variant (2 element array doubles); output-only The XY coordinates for the given station and

elevation.

StationOffset Method

Given a Northings and Eastings, returns the station, an offset, and the polar direction.

See Also | Example

Signature

object.StationOffset(Easting, Northing, Station, Offset, Direction)

object

Alignment The object or objects this property

applies to.

Easting

Double; input-only

The Easting of the coordinate to query for.

Northing

Double; input-only

The Northing of the coordinate to query for.

Station

Double; output-only

The Station of the queried point.

Offset

Double; output-only

The Offset of the queried point.

Direction

Double; output-only The direction of the queried point.

Remarks

If the Offset is to the left of the alignment, it will be negative. If the Offset is to the right of the alignment, it will be positive.

UnlockPoints Method

Unlocks a list of point numbers.

See Also | Example

Signature

object.UnlockPoints(PointString)

object

CogoPoints The object or objects this property

applies to.

PointString

String; input-only

Comma delimited string of points to be unlocked

with groups separated by hyphens.

Remarks

Typical examples of the point string format are: "1,2,3,5,10" "1-100,1000-2000"

XyToEastNorth Method

Given a AutoCAD X, Y, returns the Easting and Northing.

See Also | Example

Signature

RetVal = object.XyToEastNorth(Xy)

object

Utility The object or objects this property applies to.

Xy

Variant (3 element array of doubles); input-only

The X and Y coordinates.

RetVal

Variant (3 element array of doubles)

The coordinates in Easting and Northing.

Remarks

EastNorth must be declared as a variant and not as an array. To access the Easting and Northing, subscript the variant (var(0), var(1)). The third element (elevation) is unchanged.

XyToOffsetElevation Method (Civil Engineering Features)

Given an AutoCAD XY cooridanates, return the cross section offset and elevation.

See Also | Example

Signature

RetVal = object.XyToOffsetElevation(Xy)

object

CrossSectionBlock The object or objects this

property applies to.

 $\mathbf{X}\mathbf{y}$

Variant (2 element array of doubles); input-only

The X and Y coordinates to query for.

Retval

Variant (2 element array of doubles); output-only

The offset and elevation for the given XY

cooridnates.

XyToStationElevation Method (Civil Engineering Feature)

Given an AutoCAD XY, return the profile station and elevation.

See Also | Example

Signature

RetVal = object.XyToStationElevation(Xy)

object

<u>ProfileBlock</u> The object or objects this property

applies to.

 $\mathbf{X}\mathbf{y}$

Variant (2 element array doubles); input-only

The X and Y coordinates to guery for.

Retval

Variant (2 element array doubles); output-only

The station and elevation for the given XY

cooridnates.

Properties



Α

A The spiral's A value.

<u>Active</u> Specifies if the projects is currently loaded.

ActiveDocument Specifies the active document (drawing file).

ActiveProject Specifies the currently opened Land Desktop project

AD1 Returns AD1 which is the radius at the end of the spiral (SC).

AD2 Returns AD2 which is the radius at the start of the spiral (TS).

AlignEntities The alignment's entities collection.

<u>Alignment</u> The collection of alignment preferences for the project.

<u>Alignments</u> Gets the Alignments collection.

AngularAzimuth The format used to display angles.

<u>AngularDisplayFormat</u> The units used to display angles.

Angular Precision Specifies the angular precision for the drawing.

<u>Application</u> Gets the AeccApplication object.

Area The area of the parcel.

<u>Area2D</u> The 2D area of a Surface or Face.

Area3D The 3D area of a Surface or Face.

<u>AreaDisplayUnit</u> The units used to display areas.

<u>AreaPrecision</u> The precision used to display areas.

AreaSuffix The suffix displayed after an area.

<u>Application</u> Gets the AeccApplication object.

Author Specifies the author of the project.

<u>AutoSave</u> Returns and sets the AutoSave for Alignments, CogoPoints, DescriptionKeyFiles, and PointGroups.

<u>AverageGrade</u> Returns the BreakLines collection.

В

BasePoint Gets and sets the base point for the drawing as an XYZ.

BasePointNE Gets and sets the base point for the drawing as an ENZ.

BeginCondition Returns the BeginCondition for the Spiral.

<u>BorderBlockFileName</u> Specifies the block filename to insert as a border.

BorderBottomMargin Specifies the bottom margin for the border.

BorderLeftMargin Specifies the left margin for the border.

BorderLineWidth Specifies the line width of the border.

BorderPath Specifies the directory in which the border files are located.

BorderRightMargin Specifies the right margin for the border.

BorderStyle Specifies the border style to use for the drawing.

<u>BorderTopMargin</u> Specifies the top margin for the border.

Boundaries Returns the Boundaries collection.

Boundary Returns the boundary of the Watershed.

BreakLines Returns the BreakLines collection.

C

CCWFlag The counter-clockwise flag for the curve or parcel.

<u>CenterEasting</u> The Easting coordinate for the alignment curves or parcel curves center.

<u>CenterlineOffset</u> (Civil Engineering Feature) Returns the centerline offset of the cross section.

<u>CenterNorthing</u> The Northing coordinate for the alignment curves or parcel curves center.

CentroidEasting The easting coordinate of the parcel centroid.

<u>CentroidNorthing</u> The northing coordinate of the parcel centroid.

<u>ChordDirection</u> The direction of the chord for the curve.

<u>ChordLength</u> The chord length of the alignment or parcel curve.

<u>CivilDataFilesPath</u> Specifies the directory in which the Civil Engineering Feature data is located.

Code Returns the cross section point code or description key code.

<u>Cogo</u> The collection of cogo preferences for the project.

<u>CogoPoints</u> Gets the CogoPoints collection.

<u>Contourltems</u> Returns the Contourltems collection.

<u>ContourStyle</u> Returns and set the Contour Style object for this contour.

<u>ContourStyleName</u> Returns the Contour Style name.

<u>ContourStylesPath</u> Specifies the directory in which the contour styles are located.

<u>CoordinatePrecision</u> Specifies the coordinate precision for the drawing.

Coordinates The coordinate properties for various objects.

<u>CoordinateZone</u> Specifies the coordinate zone.

Count Gets the number of items in the collection.

Course Returns the Course of the parcel entity.

Courseln Returns the Course In of the Parcel.

CourseOut Returns the Course Out of the Parcel.

<u>CrossSection</u> (Civil Engineering Feature) The collection of cross section preferences for the project.

<u>CrossSectionBlocks</u> (Civil Engineering Feature) Gets the CrossSectionBlock object.

<u>CrossSectionPointCodes</u> (Civil Engineering Feature) Returns the cross sections point code collection.

<u>CrossSections</u> (Civil Engineering Feature) Returns the alignment cross sections collection.

<u>CrossSectionSurfaces</u> (Civil Engineering Feature) Returns the cross sections surfaces collection.

<u>CurrentAlignment</u> Sets and gets the current alignment.

<u>CurrentSurface</u> Gets / sets the current Surface.

<u>CurveCode</u> (Civil Engineering Feature) Returns the superelevation curve code.

<u>CurveLabelIncrement</u> (Civil Engineering Feature) Returns the distance between vertical curve elevation labels.

<u>CurveLength</u> (Civil Engineering Feature) Returns the Curve Length of the PVI.

<u>DatabaseScale</u> Specifies the database scale at which the horizontal axis of the drawing is displayed.

<u>DataPath</u> Specifies the directory in which the Land Desktop data files are installed.

<u>DatumElevation</u> (Civil Engineering Feature) Returns the datum elevation for the alignment profile or cross section.

Delta The delta value for a AlignCurve, AlignSpiral, or ParcelCurve.

DEMFiles Returns the DEMFiles collection.

<u>Description</u> The description properties for various objects.

<u>DescriptionFormat</u> Specifies the description key format.

DescriptionKeyFiles Gets the DescriptionKeyFiles collection.

<u>DescriptionLayer</u> Specifies the description key layer.

<u>DescriptionOverride</u> Specifies if the description for the project database point will be overridden from the Point Group.

<u>DescriptionXDRef</u> Specifies the description override for the project database point group.

<u>Direction</u> The Direction of the ProfileBlock or alignment tangent.

<u>Documents</u> Gets the Document collection.

DrainsInto Returns an array of Ids that the Watershed drains into.

<u>DrawingPath</u> Specifies the drawing path of the project.

<u>Drawings</u> Gets the Drawings collection.

<u>DrawingSetupPath</u> Specifies the directory in which the drawing setup files are located.

Easting The Easting coordinates properties for various objects.

Edges Returns the Edges collection.

<u>EGPrecision</u> (Civil Engineering Feature) Returns the existing ground precision for the alignment profile or cross section.

<u>EGProfiles</u> (Civil Engineering Feature) Gets the existing ground profiles collection.

<u>Elevation</u> The elevation property for various objects.

ElevationContours Returns the ElevationContours collection.

<u>ElevationOverride</u> Specifies if the elevation for the project database point will be overridden from the Point Group.

ElevationPrecision Specifies the elevation precision for the drawing.

<u>ElevationXDRef</u> Specifies the elevation override for the project database point group.

EndDirection The ending direction of the curve.

<u>EndEasting</u> The Easting coordinate for the end of the individual object entities.

EndingStation The object's ending station.

<u>EndNorthing</u> The Northing coordinate for the end of the individual object entity.

ExceedBoth If true, then both minimum depression settings must be met.

ExtEasting The Easting coordinate for the External point (Ext).

ExternalSecant The external secant length for the curve.

ExtNorthing The Northing coordinate for the External point (Ext).

Faces Returns the Faces collection.

FacetDeviation Gets and sets the facet deviation.

<u>FGPrecision</u>(Civil Engineering Feature) Returns the finished ground precision for the alignment profile or cross section.

FGProfiles (Civil Engineering Feature) Gets the FGProfiles collection.

<u>File</u> The filename of the file lock.

FileLocks Gets the FileLocks collection.

Files Gets the PreferencesFiles object.

<u>FirstTimeDrawingSetup</u> Selects how new drawings are setup.

<u>FirstTimeDrawingSetupFile</u> The drawing setup file loaded automatically for a new drawing.

Format The format for the PointFile.

<u>FormatsPath</u> Specifies the directory in which the point import and export formats are located.

FullDescription Returns the full description for the Cogo Point.

<u>FullName</u> The full name includes the path and file name.

G

GridEasting The grid Easting for the COGO project database point.

GridNorthing The grid Northing for the COGO project database point.

GroupName Gets / sets the group name for various objects.

Н

<u>Height</u> (Civil Engineering Feature) Returns the height of the alignment cross section.

HelpPath Specifies the directory in which the help files are installed.

ı

<u>Id</u> The identification number for the Boundary, BreakLine, ContourItem or Watershed.

<u>Inputs</u> Returns the SurfaceInputs object for the surface.

<u>IsBreakLine</u> Is the Boundary considered a BreakLine?

<u>IsNameSupported</u> Returns whether point names are supported or not.

IsVisible Returns if the Face is visible.

J

K

K The spiral's K value.

<u>Keywords</u> Specifies the keywords for the project.

L

<u>Label</u> The name of the locked data file or folder.

<u>LabelPoints</u> Returns the contour label points.

<u>LabelStyle</u> Specifies the label style override for the project database point group.

<u>LabelStyleOverride</u> Specifies if the label style for the project database point will be overridden from the Point Group.

<u>LabelStylePath</u> Specifies the directory in which the line, curve, spiral and point label style files are located.

<u>LabelStyleXDRef</u> Specifies the label style override for the project database point group.

LastUsedDwg The last used drawing.

<u>LastUsedDwgPath</u> The last used drawing path.

<u>LastUsedProj</u> The last used project.

<u>LastUsedProjPath</u> The last used project path.

<u>Latitude</u> The latitude of the point.

<u>LayerFile</u> The layer file to apply to the drawing.

<u>LayerStandard</u> The layer standard to apply to the drawing.

<u>LeftWidth</u> (Civil Engineering Feature) Returns the left width of the alignment cross section.

<u>Length</u> The length of the alignment curve, spiral or tangent or parcel curve or line.

LExt Returns the external length of the spiral.

<u>LinearDisplayUnit</u> The format used to display linear values.

<u>LinearPrecision</u> Specifies the linear precision for the drawing.

<u>LinearUnit</u> The unit used to display linear values.

<u>LockedPointNumbers</u> Returns a list of the locked points in the current project point database.

<u>LockType</u> Specifies the type of lock on the Alignment, COGO Point, or Surface.

LOffset The offset distance for the spiral.

Longitude The longitude of the point.

MaskBlockStyles Gets the Masking Block Styles collection for the drawing.

MassGroups Gets the MassGroups collection for the drawing.

<u>MaxElevation</u> Returns the Maximum Elevation of a Cross Section or Surface.

MaxFaceArea The area of the largest Face for the Surface.

MaxGrade The maximum grade for the Surface.

MaxOffset (Civil Engineering Feature) Returns the Maximum Offset of the Cross Section.

MeanElevation The mean elevation of the Surface.

MeausurementUnit The system of units used to display measurements.

MidOrdinate The mid ordinate for the curve.

MinDepressionArea The minimum depression area for all watersheds.

MinDepressionDepth The minimum depression depth for all watersheds.

MinElevation Returns the Minimum Elevation of a Cross Section or Surface.

MinFaceArea The area of the smallest Face for the Surface.

MinGrade The minimum grade for the Surface.

MinOffset (Civil Engineering Feature) Returns the Minimum Offset of the Cross Section.

MVBlockStyles Gets the Multiview Block Styles collection for the drawing.

Name Gets the name of various objects.

<u>NameOverride</u> Specifies if the name for the project database point will be overridden from the Point Group.

NameXDRef Specifies the Name override for the project database point group. The name will be taken from the external database reference (XDRef).

<u>NextPointNumber</u> Gets / sets the next sequential point number for point creation.

Normal Returns the Normal for the Face.

Northing Gets the Northing coordinates of various objects.

NorthRotation Specifies the north rotation for your drawing layout.

Number The number property for various objects.

NumberOfFaces The number of Faces for the Surface.

NumberOfPoints The number of TinPoints for the Surface.

0

ObjectID Returns the Object ID for the entity.

Offset (Civil Engineering Feature) Returns the cross section point code offset.

OffsetElevations (Civil Engineering Feature) Returns the cross section surface offset and elevations.

Outputs Returns the SurfaceOutputs object for the surface.

OverflowPoints Returns the WaterShed overflow points.

<u>OverrideDescription</u> Returns the description that has been overridden by the PointGroup.

OverrideElevation Returns the elevation that has been overridden by the PointGroup.

<u>OverrideName</u> Returns the point name that has been overridden by the PointGroup.

OverrideNew Toggles the display of the Land Desktop New dialog box.

OverrideOpen Toggles the display of the Land Desktop Open dialog box.

Owner The owner (AutoCAD login name) of the lock owner.

P

P The spiral's P value.

<u>Parcel</u> The collection of parcel preferences for the project.

<u>ParcelEntities</u> The collection of geometric entities for the Parcel.

Parcels Gets the Parcels collection.

Path Gets the path of various objects.

<u>Perimeter</u>The perimeter of the parcel.

<u>PiEasting</u> The Easting coordinate for the curve's PI.

<u>PiNorthing</u> The Northing coordinate for the curve's PI.

<u>PointFiles</u> Returns the point files collection.

<u>PointGroupNames</u> Returns the PointGroupNames collection.

<u>PointGroups</u> Gets the PointGroups collection.

<u>PointList</u> Specifies the list of project database points for the point group.

<u>PointNameSize</u> Returns maximum number of characters supported in point name.

<u>PointOnlineTolerance</u> Returns the minimum distance between a line and a point not on the line.

<u>PointTolerance</u> Returns the minimum distance between distinct points.

<u>Precision</u> The precision used to truncate area labels when a Parcel is imported to the drawing.

<u>Preferences</u> The preferences property of various objects.

<u>PreferencesPath</u> Specifies the directory in which the preference settings are located.

<u>Profile</u> Gets the vertical profile settings for the project.

ProfileBlocks (Civil Engineering Feature) Gets the ProfileBlocks object.

<u>ProfileStyles</u> Gets the ProfileStyles collection for the drawing.

<u>ProgramPath</u> Specifies the directory in which the Land Desktop is installed.

ProjectName The name of the project that the drawing belongs to.

ProjectPath Gets the project path of various objects.

<u>Projects</u> Gets the Projects collection.

PrototypeName Specifies the name of the prototype used for the project.

<u>PrototypePath</u> Specifies the directory in which the project prototype files are located.

Prototypes Gets the Prototypes collection.

PVIs Gets the PVIs collection.

Q

R

RadialDistance The radial distance of the spiral.

Radius The radius of the alignment or parcel curve.

<u>RawDescription</u> The raw description for the COGO project database point.

RevisionNumber The revision number for the Surface.

<u>RightWidth</u> (Civil Engineering Feature) Returns the right width of the alignment cross section.

<u>RotateByDescriptionParam</u> If TRUE, only the RotateDescriptionParam is rotated.

RotateByFixedFactor If TRUE, rotate by the RotateFactor.

RotateClockwise The rotation direction.

RotateDescriptionParam The description parameter for rotation.

RotateFixedFactor The rotation factor.

S

<u>ScaleByDescriptionParam</u> If TRUE, only the ScaleDescriptionParam is scaled.

<u>ScaleByDrawingScale</u> If TRUE, the drawing scale factor is applied.

<u>ScaleByFixedFactor</u> If TRUE, the fixed scale factor ScaleFactor is applied.

<u>ScaleDescriptionParam</u> The description parameter for scaling.

<u>ScaleFixedFactor</u> The fixed scale factor.

ScaleInXY If TRUE, the symbol is scaled in the XY plane

<u>ScaleInZ</u> If TRUE, the symbol is scaled in the Z axis.

<u>ScaleOnInsert</u> Specifies whether objects are automatically scaled when inserted into the drawing.

<u>SearchType</u> Returns how the Faces collection was generated.

SheetHeight Specifies the height of the sheet.

SheetWidth Specifies the width of the sheet.

<u>ShortTangent</u> Returns the short tangent length for compound and simple spirals.

ShowStartupDialog Toggles the display of the startup dialog.

<u>ShowSubfolders</u> Determines whether drawings in subfolders are included.

<u>SpeedTablesPath</u> Specifies the directory in which the speed tables are located.

SpiEasting The Easting coordinate for the spi point.

<u>SpilTangent</u> Returns the long tangent length.

SpiNorthing The Northing coordinate for the spi point.

SpiralType1 Returns the calculation method of the spiral.

<u>SpiralType2</u> Returns whether the spiral is a simple, offset, or compound spiral.

<u>StartDirection</u> The starting direction of the curve or spiral.

<u>StartEasting</u> The Easting coordinate for the beginning of the individual alignment or profile entities.

StartingStation The object's starting station.

<u>StartNorthing</u> The Northing coordinate for the beginning of the individual alignment or profile entities.

<u>Station</u> (Civil Engineering Feature) Returns the station for the cross section, cross section block, superelevation, or PVI.

StationAhead The ahead station for the station equation.

StationBack The back station for the station equation.

<u>StationElevations</u> (Civil Engineering Feature) An array of stations and elevations for the VerticalSurface

StationEquations The alignment's station equations collection.

<u>StationIncrement</u> (Civil Engineering Feature) Returns the horizontal distance between vertical grid lines.

Status The status for a surface.

<u>SuperelevationCode</u> (Civil Engineering Feature) Returns the superelevation code.

<u>Surface</u> The collection of surface preferences for the project.

<u>SurfaceName</u> (Civil Engineering Feature) The name of the EGProfile surface.

Surfaces Gets the Surfaces collection.

<u>SymbolBlock</u> Specifies the symbol block.

SymbolLayer Specifies the symbol block layer.

<u>SymbolManagerPath</u> Specifies the directory in which the symbol sets are located.

<u>SystemPath</u> Specifies the directory in which the Land Desktop is installed.

Т

<u>TangentLabelIncrement</u> (Civil Engineering Feature) Returns the distance between tangent elevation labels.

<u>TangentLength</u> The tangent length of the curve.

<u>TempPath</u> Specifies the directory in which the temporary files are stored.

<u>TextAbove</u> The text above the object.

<u>TextBelow</u> The text below the object.

<u>TextHeight</u> The height of the text in the drawing.

<u>TextOffsetAbove</u> The offset of the text above the object.

<u>TextOffsetBelow</u> The offset of the text below the object.

<u>TextSize</u> The size of the text above and below the object.

<u>ThetaExt</u> Returns the external theta for a compound spiral.

<u>Time</u> The lock creation date and time.

<u>TinPoints</u> Returns the TinPoints collection.

<u>TotalX</u> The total X value for the spiral.

<u>TotalY</u> The total Y value for the spiral.

Type The type properties for Objects.

U

<u>UpperRight</u> Returns the Upper Right Coordinates of the ProfileBlock as XY.

<u>UsedPointNumbers</u> Returns the list of used point numbers in the current project database.

<u>User</u> Gets the PreferencesUser object.

Utility Gets the Utility object.

<u>VerticalScale</u> Specifies the scale at which the vertical axis of the drawing is displayed, or the vertical scale for alignment profiles and cross sections.

<u>Volume</u> The volume of the surface.

VolumeDisplayUnit The units used to display volumes.

VolumePrecision The precision used to display volumes.

<u>VolumeSuffix</u> The suffix displayed after an volume.

W

WaterSheds Returns the WaterSheds collection.

X

Υ

Ζ

A Property

The spiral's A value.

See Also | Example

Signature

object.A

object

AlignSpiral The object or objects this property

applies to.

Α

Double; read-only The spiral's A value.

Active Property

Specifies if the projects is currently loaded.

See Also | Example

Signature

object.Active

object

Project The object or objects this property applies

to.

Active

Boolean; read-only

TRUE: The project is loaded. FALSE: The project is not loaded.

Remarks

The <u>ActiveProject</u> property of the <u>AeccApplication</u> object is the only instance of AeccProject with this property set to TRUE. This indicates that access to project owned data in external databases, such as CogoPoints, is possible.

ActiveDocument Property

Specifies the active document (drawing file).

See Also | Example

Signature

object.ActiveDocument

object

AeccApplication The object or objects this property

applies to.

ActiveDocument

Document object; read-only

Remarks

In the Preview, the ActiveDocument property returns the opened document (an AutoCAD Land Desktop drawing).

ActiveProject Property

Specifies the currently opened AutoCAD Land Desktop project.

See Also | Example

Signature

object.ActiveProject

object

AeccApplication The object or objects this property

applies to.

ActiveProject

Project object; read-only

Remarks

The ActiveProject property allows access to all project data.

AD1 Property

Returns AD1 which is the radius at the end of the spiral (SC).

See Also | Example

Signature

object.AD1

object

AlignSpiral The object or objects this property

applies to.

AD1

Double; read-only

The radius at the end of the spiral (SC).

AD2 Property

Returns AD2 which is the radius at the start of the spiral (TS).

See Also | Example

Signature

object.AD2

object

AlignSpiral The object or objects this property

applies to.

AD2

Double; read-only

The radius at the start of the spiral (TS).

Remarks

AD2 is the radius at the start of the spiral (TS = -99999.99 for infinite radius, or a value for compound spiral).

AlignEntities Property

The alignment's entities collection.

See Also | Example

Signature

object.AlignEntities

object

Alignment The object or objects this property

applies to.

AlignEntities

AlignEntities collection; read-only

The <u>AlignEntities</u> collection.

Alignment Property

The collection of alignment preferences for the project.

See Also | Example

Signature

object.Alignment

object

PreferencesProject The object or objects this

property applies to.

Alignment

<u>PreferencesAlignment</u>; read-only The <u>PreferencesAlignment</u> object.

Alignments Property

Gets the Alignments collection.

See Also | Example

Signature

object.Alignments

object

Project The object or objects this property applies

to.

Alignments

Alignments collection; read-only

The Alignments collection.

Remarks

The <u>Alignments</u> collection represents all of the <u>Alignment</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

Angular Azimuth Property

The format used to display angles.

See Also | Example

Signature

object.AngularAzimuth

object

<u>DatabasePreferences</u> The object or objects this property applies to.

AngularAzimuth

Long; read-write

0: Use Bearings

1: Use North Azimuths

2: Use South Azimuths

Angular Display Format Property

The units used to display angles.

See Also | Example

Signature

object. Angular Display Format

object

<u>DatabasePreferences</u> The object or objects this property applies to.

Angular Display Format

Long; read-write

1: Use Degrees/Minutes/Seconds

2: Use Grads

Remarks

When you type degrees at AutoCAD Land Desktop command prompts, type them in the decimal format indicated (DD.MMSS). Use a period (.) between the degree value and the minutes and seconds. For example, to enter 67°45'15" type the value as 67.4515. If you use bearings, then type the bearing quadrant first, and then the angle in degrees.

Angular Precision Property

Specifies the angular precision for the drawing.

See Also | Example

Signature

object.AngularPrecision

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

AngularPrecision

Long; read-write

The number of decimals to the right of the decimal

point.

Remarks

This is used to display minutes and seconds.

The precision settings are used only for labeling and listing values, not actual computations. AutoCAD Land Desktop commands always calculate all numbers up to the highest internal precision.

Application Property

Gets the AeccApplication object.

See Also | Example

Signature

object.Application

object

All objects The object or objects this property

applies to.

Application

Application object, read-only

Remarks

The <u>Application</u> object represents the AutoCAD Land Desktop, and provides the means to navigate to the top of the object hierarchy. All objects in the object model implement the Application property.

Area Property

The area of the parcel.

See Also | Example

Signature

object.Area

object

Parcel The object or objects this property applies to.

Area

Double; read-only

The area of the parcel.

Area2D Property

The 2D area of a Surface or Face.

See Also | Example

Signature: Overview

l Face

1 Surface

The 2D area of a Face.

Signature

object.Area2D

object

Face The object or objects this property applies to.

Area2D

Double; read-only The 2D face area.

Remarks

The 2D face area is the apparent triangle area if you look at the face from plan view.

The 2D area of a Surface.

Signature

object.Area2D

object

Surface

The object or objects this property applies to.

Area2D

Double; read-only The 2D surface area.

Remarks

The 2D surface area is the apparent surface area if you look at the surface from plan view. It is obtained by projecting the visible triangles onto the XY plane along the Z axis and summing the areas of the triangles.

If areas on the surface are hidden within boundaries, then these areas are not included in the surface area.

This is an Extended Statistic.

Area3D Property

The 3D area of a Surface or Face.

See Also | Example

Signature: Overview

l Face

1 Surface

The 3D area of the Face.

Signature

object.Area3D

object

Face The object or objects this property applies to.

Area3D

Double; read-only The 3D face area.

Remarks

The 3D face area is the true area of the triangle.

The 3D area of the Surface.

Signature

object.Area3D

object

Surface

The object or objects this property applies to.

Area3D

Double; read-only The 3D surface area.

Remarks

The 3D surface area is the true area of the surface and accounts for variations in the surface elevation. The 3D area is the sum of the areas of each of the visible triangles in the surface without projecting the triangles. The greater the variation in elevations, the more the 3D area will differ from the 2D area.

An example of the difference between 2D and 3D areas is a building pad. The 2D area is the area that you can see from only the top and doesn't vary with the height of the building pad. The 3D surface area of the building pad is the sum of the areas of each face of the building pad, which would become larger the higher the building pad becomes.

This is an Extended Statistic.

AreaDisplayUnit Property

The units used to display areas.

See Also | Example

Signature

object.AreaDisplayUnit

object

DatabasePreferences The object or objects this

property applies to.

Area Display Unit

AecBuiltInUnit enum; read-write

aecUnitSquareInch: Square inch unit

aecUnitSquareFoot: Square foot unit

aecUnitSquareYard: Square yard unit

aecUnitSquareMil: Square millimeter unit

aecUnitSquareCentimeter: Square centimeter unit

aecUnitSquareDecimeter: Square decimeter unit

aecUnitSquareMeters: Square meter unit

Remarks

The AreaDisplayUnit supports more units than are shown above. The units shown above are those that are supported by the AutoCAD Land Desktop.

AreaPrecision Property

The precision used to display areas.

See Also | Example

Signature

object.AreaDisplayUnit

object

DatabasePreferences The object or objects this

property applies to.

AreaPrecision

Long; read-write

The display precision used for areas.

Remarks

The precision settings are used only for labeling and listing values, not actual computations. AutoCAD Land Desktop commands always calculate all numbers up to the highest internal precision.

AreaSuffix Property

The suffix displayed after an area.

See Also | Example

Signature

object.AreaDisplayUnit

object

DatabasePreferences The object or objects this

property applies to.

AreaPrecision

String; read-write

The suffix displayed after an area.

Remarks

The default value is "Sq.Ft." or "m2", depending on the value of the MeasurementUnit property. The value must 255 characters or less in length.

Author Property

Specifies the author of the project.

See Also | Example

Signature

object.Author

object

Project The object or objects this property applies

to.

Author

String; read-write

The author of the project.

AutoSave Property

Returns and sets the AutoSave for Alignments, CogoPoints, DescriptionKeyFiles, and PointGroups.

See Also | Example

Signature: Overview

- 1 Alignments
- 1 CogoPoints
- 1 <u>DescriptionKeyFiles</u>
- l PointGroups

Returns and sets the AutoSave.

Signature

object.AutoSave

object

Alignments The object or objects this property

applies to.

AutoSave

Boolean; read-write

TRUE: AutoSave is enabled FALSE: AutoSave is disabled

Remarks

The initial value for this property is TRUE. Each addition or change is written to the Alignments file immediately. If this property is set to FALSE, the <u>Save</u> method must be called to write additions or changes to the Alignment file. The <u>Add</u> method for <u>Alignments</u> is always executed immediately, regardless of the AutoSave setting.

Returns and sets the AutoSave.

Signature

object.AutoSave

object

CogoPoints

The object or objects this property applies to.

AutoSave

Boolean; read-write

TRUE: AutoSave is enabled FALSE: AutoSave is disabled

Remarks

The initial value for this property is TRUE. Each addition or change is written to the CogoPoints file immediately. If this property is set to FALSE, the <u>Save</u> method must be called to write additions or changes to the CogoPoints file. The <u>Add</u> method for <u>CogoPoints</u> is always executed immediately, regardless of the AutoSave setting.

Returns and sets the AutoSave.

Signature

object.AutoSave

object

<u>DescriptionKeyFiles</u>

The object or objects this property applies to.

AutoSave

Boolean; read-write

TRUE: AutoSave is enabled FALSE: AutoSave is disabled

Remarks

The initial value for this property is TRUE. Each addition or change is written to the DescriptionKeyFiles file immediately. If this property is set to FALSE, the <u>Save</u> method must be called to write additions or changes to the DescriptionKeyFiles file. The <u>Add</u> method for <u>DescriptionKeyFiles</u> and <u>DescriptionKeyFiles</u> are always executed immediately, regardless of the AutoSave setting.

Returns and sets the AutoSave.

Signature

object.AutoSave

object

PointGroups

The object or objects this property applies to.

AutoSave

Boolean; read-write

TRUE: AutoSave is enabled FALSE: AutoSave is disabled

Remarks

The initial value for this property is TRUE. Each addition or change is written to the PointGroups collection immediately. If this property is set to FALSE, the <u>Save</u> method must be called to write additions or changes to the PointGroups collection. The <u>Add</u> method for <u>PointGroups</u> is always

executed immediately, regardless of the AutoSave setting.

AverageGrade Property

The average grade for the Surface.

See Also | Example

Signature

object. Average Grade

object

Surface The object or objects this property applies

to.

AverageGrade

Double; read-only

The average grade for the surface.

Remarks

This is an Extended Statistic.

BasePoint Property

Gets and sets the base point for the drawing as an XY.

See Also | Example

Signature

object.BasePoint

object

DatabasePreferences The object or objects this

property applies to.

BasePoint

Variant (3 element array of doubles); read-write

The base point in WCS X, Y, Z.

Remarks

Set this value to indicate the WCS X, Y, Z that corresponds to the value of the <u>BasePointNE</u> property. In the default coordinate system X,Y is 0,0 and the Northing/Easting is 0,0.

BasePointNE Property

Gets and sets the base point for the drawing as an ENZ.

See Also | Example

Signature

object.BasePointNE

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

BasePointNE

Variant (3 element array of doubles); read-write

The base point in WCS X, Y, Z.

Remarks

Set this value to indicate the Easting, Northing and elevation that corresponds to the value of the <u>BasePoint</u> property. In the default coordinate system X,Y is 0,0 and the Northing/Easting is 0,0.

BeginCondition Property

Returns the BeginCondition for the Spiral.

See Also | Example

Signature

object.BeginCondition

object

AlignSpiral The object or objects this property

applies to.

Begin Condition

eAeccSpiralBegin enum; read-only

kSC: Spiral begins on the SC point

kTS: Spiral begins on the TS point

Remarks

The BeginCondition determines whether the spiral begins on its SC or TS point.

BorderBlockFilename Property

Specifies the block filename to insert as a border.

See Also | Example

Signature

object.BorderBlockFilename

object

<u>DatabasePreferences</u> The object or objects this property applies to.

BorderBlockFilename

String; read-write The border block filename.

Remarks

The block names comply with the following naming conventions: pf_{-} are plan borders for drawings that use feet as units pm_{-} are plan borders for drawings that use meters as units df_{-} are detail borders for drawings that use feet as units dm_{-} are detail borders for drawings that use meters as units

BorderBottomMargin Property

Specifies the bottom margin for the border.

See Also | Example

Signature

object.BorderBottomMargin

object

<u>DatabasePreferences</u> The object or objects this property applies to.

BorderBottomMargin

Double; read-write
The border bottom margin.

BorderLeftMargin Property

Specifies the left margin for the border.

See Also | Example

Signature

object.BorderLeftMargin

object

DatabasePreferences The object or objects this

property applies to.

BorderLeftMargin

Double; read-write

The border left margin.

BorderLineWidth Property

Specifies the line width of the border.

See Also | Example

Signature

object.BorderLineWidth

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

BorderLineWidth

Double; read-write The border line width.

BorderPath Property

Specifies the directory in which the border files are located.

See Also | Example

Signature

object.BorderPath

object

PreferencesFiles The object or objects this property

applies to.

BorderPath

String; read-write

The drive and path for the drawing setup border

.dwg files.

Remarks

The project path is C:\My DataPath\borders. By default, the data directory is C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data. You must exit AutoCAD Land Desktop and restart for this change to take effect.

BorderRightMargin Property

Specifies the right margin for the border.

See Also | Example

Signature

object. Border Right Margin

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

BorderRightMargin

Double; read-write

The border right margin.

BorderStyle Property

Specifies the border style to use for the drawing.

See Also | Example

Signature

object.BorderStyle

object

DatabasePreferences The object or objects this

property applies to.

Border Style

eAeccBorderStyle enum; read-write

The type of the Border.

kBorderStyleLine: Insert a polyline

kBorderStyleUnscaledBlock: Insert a block at 1:1

kBorderStyleScaledBlock: Insert a block using

<u>DatabaseScale</u>

kBorderStyleNone: No border

BorderTopMargin Property

Specifies the top margin for the border.

See Also | Example

Signature

object.BorderTopMargin

object

DatabasePreferences The object or objects this

property applies to.

BorderTopMargin

Double; read-write

The border top margin.

Boundaries Property

Returns the Boundaries collection.

See Also | Example

Signature

object.Boundaries

object

<u>SurfaceInputs</u> The object or objects this property

applies to.

Boundaries

Boundaries collection; read-only

The **Boundaries** collection.

Boundary Property

Returns the boundary of the Watershed.

See Also | Example

Signature

object.Boundary

object

WaterShed The object or objects this property

applies to.

Boundary

Variant (array of doubles); read-only

An array of doubles representing the vertices of the

boundary for the WaterShed.

BreakLines Property

Returns the BreakLines collection.

See Also | Example

Signature

object.BreakLines

object

<u>SurfaceInputs</u> The object or objects this property

applies to.

BreakLines

BreakLines collection; read-only

The **BreakLines** collection.

CCWFlag Property

The counter-clockwise flag for the curve or parcel.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The counter-clockwise flag for the curve.

Signature

object.CCWFlag

object

AlignCurve The object or objects this property

applies to.

CCWFlag

Boolean; read-only

TRUE: The curve is counter-clockwise.

FALSE: The curve is clockwise.

The counter-clockwise flag for the parcel.

Signature

object.CCWFlag

object

ParcelCurve

The object or objects this property applies to.

CCWFlag

Boolean; read-only TRUE: The curve is counter-clockwise.

FALSE: The curve is clockwise.

CenterEasting Property

The Easting coordinate for the alignment curves or parcel curves centroid.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The Easting coordinate for the curve's centroid.

Signature

object.CenterEasting

object

AlignCurve The object or objects this property

applies to.

CenterEasting

Double; read-only

The east coordinate for the center of the curve.

The Easting coordinate for the parcel's centroid.

Signature

object.CenterEasting

object

ParcelCurve

The object or objects this property applies to.

CenterEasting

Double; read-only
The east coordinate for the center of the curve.

CenterlineOffset Property (Civil Engineering Feature)

Returns the centerline offset of the cross section.

See Also | Example

Signature

object.CenterlineOffset

object

CrossSectionBlock The object or objects this

property applies to.

CenterlineOffset

Double; read-only

The centerline offset of the cross section.

CenterNorthing Property

The Northing coordinate for the alignment curves or parcel curves center.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The Northing coordinate for the curve's center.

Signature

object.CenterNorthing

object

AlignCurve The object or objects this property

applies to.

CenterNorthing

Double; read-only

The north coordinate for the center of the curve.

The Northing coordinate for the parcel's center.

Signature

object.CenterNorthing

object

ParcelCurve

The object or objects this property applies to.

CenterNorthing

Double; read-only
The north coordinate for the center of the curve.

CentroidEasting Property

Returns the centroid easting of the Parcel.

See Also | Example

Signature

object.CentroidEasting

object

Parcel The object or objects this property applies to.

CentroidEasting

Double; read-only

The parcel's centroid easting value.

CentroidNorthing Property

The northing coordinate of the parcel centroid.

See Also | Example

Signature

object.CentroidNorthing

object

Parcel The object or objects this property applies to.

CentroidNorthing

Double; read-only

The parcel's centroid northing value.

ChordDirection Property

The direction of the chord for the curve.

See Also | Example

Signature

object.ChordDirection

object

AlignCurve The object or objects this property

applies to.

ChordDirection

Double; read-only

The angle of the chord for the curve. The angle is in radians and measured counter-clockwise from the X

axis (East).

ChordLength Property

The chord length of the alignment or parcel curve.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The chord length of the alignment curve.

Signature

object.ChordLength

object

AlignCurve The object or objects this property

applies to.

ChordLength

Double; read-only

The chord length of the curve.

The chord length of the parcel curve.

Signature

object.ChordLength

object

ParcelCurve

The object or objects this property applies to.

ChordLength

Double; read-only
The chord length of the curve.

CivilDataFilesPath Property

Specifies the directory in which the civil engineering feature data is located.

See Also | Example

Signature

object.CivilDataFilesPath

object

<u>PreferencesFiles</u> The object or objects this property applies to.

CivilDataFilesPath

String; read-write

The drive and path for the civil enginering feature

data files.

Remarks

By default, this path is C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data, the same as the default path for the AutoCAD Land Desktop data path. You must exit AutoCAD Land Desktop and restart for this change to take effect.

Code Property

Returns the cross section point code or description key code.

See Also | Example

Signature: Overview

1 <u>CrossSectionPointCode</u> (Civil Engineering Feature)

1 DescriptionKey

Returns the cross section point code.

Signature

object.Code

object

<u>CrossSectionPointCode</u> The object or objects this

property applies to.

Code

Integer; read-only

Returns the cross section point code.

Specifies the description key syntax.

Signature

object.Code

object

DescriptionKey

The object or objects this property applies to.

Code

String; read-only.

Returns the code for a description key, not including

the path.

Cogo Property

The collection of cogo preferences for the project.

See Also | Example

Signature

object.Cogo

object

PreferencesProject The object or objects this

property applies to.

Cogo

<u>PreferencesCogo</u>; read-only The <u>PreferencesCogo</u> object.

CogoPoints Property

Gets the CogoPoints collection.

See Also | Example

Signature

object.CogoPoints

object

Project The object or objects this property applies

to.

CogoPoints

CogoPoints collection; read-only

The **CogoPoints** collection.

Remarks

The <u>CogoPoints</u> collection represents all of the <u>CogoPoint</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

ContourItems Property

Returns the ContourItems collection.

See Also | Example

Signature

object.ContourItems

object

<u>SurfaceInputs</u> The object or objects this property

applies to.

ContourItems

ContourItems collection; read-only

The **ContourItems** collection.

ContourStyle Property

Returns and set the Contour Style object for this contour.

See Also | Example

Signature

object.ContourStyle

object

AeccContour The object or objects this property

applies to.

ContourStyle

AeccContourStyle object; read-write A contour style from the dictionary.

Remarks

You can use this property to get the contour style currently applied to this contour, or apply a different style.

The Contour Style object represents an entry in the dictionary named "AECC_CONTOUR_STYLES". The Contour Style object exposes a Name property, which is the keyword. You can iterate through the dictionary as a collection to determine the contour styles available in the drawing.

ContourStyleName Property

Returns the Contour Style

See Also | Example

Signature

object.ContourStyleName

object

AeccContour The object or objects this property

applies to.

Contour Style Name

String, read-write

Returns the Contour Style name.

ContourStylesPath Property

Specifies the directory in which the contour styles are located.

See Also | Example

Signature

object.ContourStylesPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

ContourStylesPath

String; read-write

The drive and path for the contour style files.

Remarks

By default, the project path is C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data\Contours. You must exit AutoCAD Land Desktop and restart for this change to take effect.

CoordinatePrecision Property

Specifies the coordinate precision for the drawing.

See Also | Example

Signature

object.CoordinatePrecision

object

<u>DatabasePreferences</u> The object or objects this property applies to.

CoordinatePrecision

Long; read-write

The number of decimals to the right of the decimal point.

Remarks

This is used to display all Northing/Easting coordinate information.

The precision settings are used only for labeling and listing values, not actual computations. AutoCAD Land Desktop commands always calculate all numbers up to the highest internal precision.

Coordinates Property

The coordinate properties for various objects.

See Also | Example

Signature: Overview

- l AeccContour
- l <u>Boundary</u>
- 1 BreakLine
- l CogoPoint
- 1 ContourItem
- l CrossSectionBlock (Civil Engineering Feature)
- l <u>Edge</u>
- 1 ElevationContour
- l Face
- l <u>ProfileBlock</u> (Civil Engineering Feature)

Returns the vertices of the Contour in XYZ as doubles.

Signature

object.Coordinates

object

AeccContour The object or objects this property

applies to.

Coordinates

Variant (three-element array of doubles); read-write

The array of points.

Returns the vertices of the Boundary in ENZ as doubles.

Signature

object.Coordinates

object

Boundary

The object or objects this property applies to.

Coordinates

Variant (three-element array of doubles); read-only

The array of points.

Returns the vertices of the BreakLine in ENZ as doubles

Signature

object.Coordinates

object

BreakLine

The object or objects this property applies to.

Coordinates

Variant (three-element array of doubles); read-only

The array of points.

Remarks

To modify the points for a BreakLine, you must deleted the BreakLine and add a new one.

Returns the point coordinates in XYZ as doubles.

Signature

object.Coordinates

object

CogoPoint

The object or objects this property applies to.

Coordinates

Variant: (3 element array of doubles); read-only

The point coordinate

Returns the vertices of the ContourItem in ENZ as doubles.

Signature

object.Coordinates

object

ContourItem

The object or objects this property applies to.

Coordinates

Variant (three-element array of doubles); read-only

The vertices of the contour.

Returns the alignment cross section coordinates.

Signature

object.Coordinates

object

CrossSectionBlock

The object or objects this property applies to.

Coordinates

Variant (two-element array of doubles); read-only

The array of X and Y values.

Remarks

The coordinates represent the AutoCAD X and Y position of the alignment cross section where the user selected the bottom insertion point.

Returns the vertices of the Edge in ENZ as doubles.

Signature

object.Coordinates

object

Edge

The object or objects this property applies to.

Coordinates

Variant (6 element array of doubles); read-only The point coordinate for the start and end of the

edge

Returns the vertices of the ElevationContour in ENZ as doubles.

Signature

object.Coordinates Points

object

ElevationContour

The object or objects this property applies to.

Coordinates

Variant (array of doubles); read-only

Contains an array of doubles in the format of

Easting, Northing, Elevation, etc.

Returns the vertices of the face in ENZ as doubles.

Signature

object.Coordinates

object

Face

The object or objects this property applies to.

Coordinates

Variant (9 element array of doubles); read-only

The point coordinates for the three points of the face

triangle.

Remarks

Returns the coordinates for the face. The vertices are numbered 0, 1, and 2. They are returned in counter-clockwise order.

Returns the Coordinates of the Alignment Profile as AutoCAD X and Y values.

Signature

object.Coordinates

object

ProfileBlock

The object or objects this property applies to.

Coordinates

Variant (two-element array of doubles); read-only The array of X and Y values.

Remarks

The coordinates represent the AutoCAD X and Y position of the lower left corner of the alignment profile where the user selected the starting point.

CoordinateZone Property

Specifies the coordinate zone.

See Also | Example

Signature: Overview

1 DatabasePreferences

1 DEMFile

Specifies the preferences coordinate zone.

Signature

object.CoordinateZone

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

CoordinateZone

String; read-write

The coordinate system syntax for the zone.

Remarks

Drawings in a project can have the same or different coordinate zones assigned to them. To work in real-world coordinates, you must establish a current zone for the drawing.

The coordinate zone code for the DEM file.

Signature

object.CoordinateZone

object

DEMFile

The object or objects this property applies to.

CoordinateZone

String; read-only

The coordinate zone code for the DEM file.

Remarks

The CoordinateZone property is optional, and can only be set with the Add method of the DEMFiles collection.

Count Property

Gets the number of items in the collection.

See Also | Example

Signature

object.Count

object

All Collections The object or objects this property

applies to.

Count

Long; read-only

The number of items in the collection.

Course Property

Returns the Course of the parcel entity.

See Also | Example

Signature

object.Course

object

ParcelEntity, ParcelCurve, ParcelLine

The object or objects this property applies to.

Course

Double; read-only

The course In for the parcel. The course angle is in radians and measured counter-clockwise from the ${\sf X}$

axis (East).

CourseIn Property

Returns the Course In of the Parcel.

See Also | Example

Signature

object.CourseIn

object

ParcelCurve The object or objects this property

applies to.

CourseIn

Double; read-only

The course In for the parcel. The course angle is in radians and measured counter-clockwise from the X

axis (East).

CourseOut Property

Returns the Course Out of the Parcel.

See Also | Example

Signature

object.CourseOut

object

ParcelCurve The object or objects this property

applies to.

CourseOut

Double; read-only

The course In for the parcel. The course angle is in radians and measured counter-clockwise from the X

axis (East).

CrossSection Property (Civil Engineering Feature)

The collection of cross section preferences for the project.

See Also | Example

Signature

object.CrossSection

object

PreferencesProject The object or objects this

property applies to.

CrossSection

<u>PreferencesCrossSection</u>; read-only The <u>PreferencesCrossSection</u> object.

CrossSectionBlocks Property

Gets the CrossSectionBlocks object.

See Also | Example

Signature

object. Cross Section Blocks

object

<u>Document</u> The object or objects this property applies to.

CrossSectionBlocks

<u>CrossSectionBlocks</u> object; read-only The <u>CrossSectionBlocks</u> object.

CrossSectionPointCodes Property (Civil Engineering Feature)

Returns the cross sections point codes collection

See Also | Example

Signature

object. Cross Section Point Codes

object

<u>CrossSection</u> The object or objects this property applies to.

CrossSectionPointCodes

<u>CrossSectionPointCodes</u> collection; read-only The <u>CrossSectionPointCodes</u> collection.

CrossSections Property (Civil Engineering Feature)

Gets the alignment cross sections collection.

See Also | Example

Signature

object.CrossSections

object

Alignment The object or objects this property

applies to.

CrossSections

CrossSections collection; read-only

The **CrossSections** collection.

CrossSectionSurfaces Property (Civil Engineering Feature)

Returns the cross sections surfaces collection

See Also | Example

Signature

object.CrossSectionSurfaces

object

<u>CrossSection</u> The object or objects this property applies to.

CrossSectionSurfaces

<u>CrossSectionSurfaces</u> collection; read-only The <u>CrossSectionSurfaces</u> collection.

CurrentAlignment Property

Sets and gets the current alignment.

See Also | Example

Signature

object.CurrentAlignment

object

Alignments The object or objects this property

applies to.

CurrentAlignment

String; read-write

The current alignment name.

Remarks

All alignments initially are loaded with read only permission. Only the current alignment can have write permissions. However, another user could have all ready locked that alignment. If this happens, read permissions will be granted for the current alignment.

By changing the current alignment, you are a) locking / unlocking alignments and b) setting the current alignment in the AutoCAD Land Desktop menus.

Setting CurrentAlignment to an empty string removes the current alignment status.

CurrentSurface Property

Gets / sets the current Surface.

See Also | Example

Signature

object.CurrentSurface

object

Surfaces The object or objects this property applies

to.

CurrentSurface

String; read-write

The current surface set in project.

Remarks

Only one surface can be considered current in project. This surface is both opened and locked.

CurveCode Property (Civil Engineering Feature)

Returns the superelevation curve code.

See Also | Example

Signature

object.CurveCode

object

Superelevation The object or objects this property

applies to.

CurveCode

eAeccSectionSECurveCode; read-only

kSERightHandCurve: Right hand curve

kSELeftHandCurve: Left hand curve

CurveLabelIncrement Property (Civil Engineering Feature)

Returns the distance between vertical curve elevation labels.

See Also | Example

Signature

object.CurveLabelIncrement

object

<u>ProfileBlock</u> The object or objects this property applies to.

CurveLabelIncrement

Double; read-only
The Curve Label increment for the Profile.

CurveLength Property (Civil Engineering Feature)

Returns the Curve Length of the PVI.

See Also | Example

Signature

object.CurveLength

object

PVI The object or objects this property applies to.

CurveLength

Double; read-only

The Curve Length for the PVI.

DatabaseScale Property

Specifies the database scale at which the horizontal axis of the drawing is displayed.

See Also | Example

Signature

object.DatabaseScale

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

DatabaseScale

Double; read-write

The database (horizontal) scale. For example, for a

1:200 scale, type 200.

Remarks

The database (horizontal) scale controls the size of annotation placed in the drawing, including text, scaled blocks and special lines. It does not affect the line lengths or point coordinates because they are always defined in real world coordinates, not to any scale. Neither does it affect any design data in the drawing or project files.

If you change the DatabaseScale property in the middle of a drawing session, then any annotation added subsequently is scaled accordingly.

DataPath Property

Specifies the directory in which the AutoCAD Land Desktop data files are installed.

See Also | Example

Signature

object.DataPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

DataPath

String; read-only

The drive and path for the data files.

Remarks

This value is read-only and cannot be changed.

DatumElevation Property (Civil Engineering Feature)

Returns the datum elevation for the alignment profile or cross section.

See Also | Example

Signature: Overview

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 ProfileBlock (Civil Engineering Feature)

Returns the datum elevation for the alignment cross section.

Signature

object.DatumElevation

object

CrossSectionBlock The object or objects this

property applies to.

DatumElevation

Double; read-only

The datum elevation for the alignment cross section.

Returns the datum elevation for the alignment profile.

Signature

object.DatumElevation

object

ProfileBlock

The object or objects this property applies to.

DatumElevation

Double; read-only The datum elevation for the alignment profile.

Delta Property

The delta value for a AlignCurve, AlignSpiral, or ParcelCurve.

See Also | Example

Signature: Overview

1 AlignCurve, AlignSpiral

1 ParcelCurve

The delta value for the alignment curve or spiral.

Signature

object.Delta

object

AlignCurve, AlignSpiral

The object or objects this property applies to.

Delta

Double; read-only

The internal (included) angle for the curve in

radians.

The delta value for the parcel curve.

Signature

object.Delta

object

ParcelCurve

The object or objects this property applies to.

Delta

Double; read-only The internal (included) angle for the curve in

radians.

DEMFiles Property

Returns the DEMFiles collection.

See Also | Example

Signature

object.DEMFiles

object

SurfaceInputs The object or objects this property

applies to.

DEMFiles

DEMFiles collection; read-only

The **DEMFiles** collection.

Description Property

The description properties for various objects.

See Also | Example

Signature: Overview

- l AeccPoint
- l <u>Alignment</u>
- l **Boundary**
- 1 BreakLine
- 1 CrossSectionPointCode (Civil Engineering Feature)
- l PointGroup
- l Project
- l Prototype
- l <u>Surface</u>

The description for the point object.

Signature

object.Description

object

AeccPoint The object or objects this property

applies to.

Description

String; read-write

The description for the point object.

The alignment description.

Signature

object.Description

object

Alignment

The object or objects this property applies to.

Description

String; read-write

The description for the alignment.

The description for the Boundary.

Signature

object.Description

object

Boundary

The object or objects this property applies to.

Description

String: read-write

The description of the boundary.

The description for the BreakLine

Signature

object.Description

object

BreakLine

The object or objects this property applies to.

Description

String; read-write

The description for the BreakLine.

The description for the cross sections point code.

Signature

object.Description

object

CrossSectionPointCode

The object or objects this property applies to.

Description

String; read-only

The description for the point code.

Specifies the description override for the project database point group.

Signature

object.Description

object

PointGroup

The object or objects this property applies to.

Description

String; read-write

Remarks

If the Description property is set, then the <u>DescriptionXDRef</u> property is cleared.

Specifies the prototype used for the project.

Signature

object.Description

object

Project

The object or objects this property applies to.

Prototype

String; read-write

The prototype used for this project.

The prototype description.

Signature

object.Description

object

Prototype

The object or objects this property applies to.

Description

String; read-write

The prototype description.

Remarks

Use this property to change the description of a prototype.

The description for the Surface.

Signature

object.Description

object

Surface

The object or objects this property applies to.

Description

String; read-write

The description for the surface.

Remarks

The Surface description can be up to 255 characters.

DescriptionFormat Property

Specifies the description key format.

See Also | Example

Signature

object.DescriptionFormat

object

<u>DescriptionKey</u> The object or objects this property

applies to.

DescriptionFormat

String; read-write

Returns and sets the description format.

DescriptionKeyFiles Property

Gets the DescriptionKeyFiles collection.

See Also | Example

Signature

object.DescriptionKeyFiles

object

<u>Project</u> The object or objects this property applies to.

DescriptionKeyFiles

<u>DescriptionKeyFiles</u> collection; read-only The <u>DescriptionKeyFiles</u> collection.

Remarks

The <u>DescriptionKeyFiles</u> collection represents all of the <u>DescriptionKeyFiles</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

DescriptionLayer Property

Specifies the description key layer.

See Also | Example

Signature

object.DescriptionLayer

object

DescriptionKey The object or objects this property

applies to.

DescriptionLayer

String; read-write

Returns and sets the description layer.

DescriptionOverride Property

Specifies if the description for the project database point will be overridden from the Point Group.

See Also | Example

Signature

object.DescriptionOverride

object

PointGroup The object or objects this property applies to.

DescriptionOverride

Boolean; read-write

TRUE: The Cogo project point description will be

set by the Point Group.

FALSE: The Cogo project point description will not

be set by the Point Group.

Remarks

When you set this value to FALSE, the <u>Description</u> property will be cleared.

DescriptionXDRef Property

Specifies the description override for the project database point group.

See Also | Example

Signature

object.DescriptionXDRef

object

PointGroup The object or objects this property

applies to.

Description XDRef

String; read-write

The name of the external database reference.

Remarks

The description will be taken from the external database reference (XDRef). If the DescriptionXDRef property is set, then the <u>Description</u> property is cleared.

Direction Property

The Direction of the ProfileBlock or alignment tangent.

See Also | Example

Signature: Overview

l <u>AlignTangent</u>

1 ProfileBlock (Civil Engineering Feature)

The direction of the alignment tangent.

Signature

object.Direction

object

AlignTangent The object or objects this property

applies to.

Direction

Double; read-only

The polar direction of the tangent.

The direction of the alignment profile.

Signature

object.Direction

object

ProfileBlock

The object or objects this property applies to.

Direction

String; read-only
The direction of the alignment profile.

Documents Property

Gets the Document collection.

See Also | Example

Signature

object.Documents

object

AeccApplication The object or objects this property

applies to.

Documents

Documents collection; read-only

Remarks

The <u>Documents</u> collection allows you to access all the documents, or drawings, in the current AutoCAD Land Desktop session.

DrainsInto Property

Returns an array of Ids that the Watershed drains into.

See Also | Example

Signature

object.DrainsInto

object

WaterShed The object or objects this property

applies to.

DrainsInto

Variant (array of longs); read-only

An array of WaterShed Ids that the current

WaterShed drains into.

DrawingPath Property

Specifies the drawing path of the project.

See Also | Example

Signature

object.DrawingPath

object

Project The object or objects this property applies

to.

DrawingPath

String; read-write

The drawing path used by the project.

Remarks

The initial value of this property is empty - the default drawing path to save drawings in is the \dwg subfolder.

Drawings Property

Gets the Drawings collection.

See Also | Example

Signature

object.Drawings

object

Project The object or objects this property applies

to.

Drawings

Drawings collection; read-only

The **Drawings** collection.

Remarks

The <u>Drawings</u> collection represents all drawings in the <u>DrawingPath</u> for this project.

DrawingSetupPath Property

Specifies the directory in which the drawing setup files are located.

See Also | Example

Signature

object.DrawingSetupPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

DrawingSetupPath

String; read-write

The drive and path for the drawing setup files.

Remarks

By default, this path is C:\Documents and Settings\All Users\Application Data\AutoCAD Land Desktop <version number>\<release number>\Data\setup. You must exit AutoCAD Land Desktop and restart for this change to take effect.

Easting Property

The Easting coordinates properties for various objects.

See Also | Example

Signature: Overview

1 AeccPoint

l CogoPoint

1 TinPoint

The Easting coordinate of the point.

Signature

object.Easting

object

AeccPoint The object or objects this property

applies to.

Easting

Double; read-write

The Easting coordinate of the point.

Remarks

The Easting coordinate is derived from the WCS X coordinate with the BasePoint and NorthRotation applied.

The east coordinate for the point.

Signature

object.Easting

object

CogoPoint

The object or objects this property applies to.

Easting

Double; read-write

The Easting coordinate for the TinPoint.

Signature

object.Easting

object

TinPoint

The object or objects this property applies to.

Easting

Double; read-only

The Easting coordinate.

Edges Property

Returns the Edges collection.

See Also | Example

Signature

object.Edges

object

SurfaceOutputs The object or objects this property

applies to.

Edges

Edges collection; read-only

The **Edges** collection.

EGPrecision Property (Civil Engineering Feature)

Returns the existing ground precision for the alignment profile or cross section.

See Also | Example

Signature: Overview

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 <u>ProfileBlock</u> (Civil Engineering Feature)

Returns the existing ground precision for the alignment cross section.

Signature

object.EGPrecision

object

CrossSectionBlock The object or objects this

property applies to.

EGPrecision

Integer; read-only

The existing ground precision for the alignment

cross section.

Returns the existing ground precision for the alignment profile.

Signature

object.EGPrecision

object

ProfileBlock

The object or objects this property applies to.

EGPrecision

Integer; read-only

The existing ground precision for the alignment

profile.

EGProfiles Property (Civil Engineering Feature)

Gets the existing ground profiles collection.

See Also | Example

Signature

object.EGProfiles

object

Alignment The object or objects this property

applies to.

EGProfiles

EGProfiles collection; read-only

The **EGProfiles** collection.

Elevation Property

The elevation property for various objects.

See Also | Example

Signature: Overview

- l AeccContour
- l AeccPoint
- l CogoPoint
- 1 <u>CrossSectionPointCode</u> (Civil Engineering Feature)
- 1 ElevationContours
- l PointGroup
- 1 PVI (Civil Engineering Feature)
- l TinPoint

The elevation of the AeccContour.

Signature

object.Elevation

object

AeccContour The object or objects this property

applies to.

Elevation

Double; read-write

The elevation of the contour.

The elevation of the AeccPoint.

Signature

object.Elevation

object

AeccPoint

The object or objects this property applies to.

Elevation

Double; read-write

The elevation of the point.

Remarks

The Elevation property is always the same as the WCS Z coordinate.

The elevation for the CogoPoints.

Signature

object.Elevation

object

CogoPoint

The object or objects this property applies to.

Elevation

Double; read-write

Remarks

The Elevation property is the raw elevation. PointGroup Overrides will not be applied to this property.

The elevation for the cross section point code.

Signature

object.Elevation

object

<u>CrossSectionPointCode</u>

The object or objects this property applies to.

Elevation

Double; read-only

The elevation for the cross section point code.

The elevation for the ElevationContours collection.

Signature

object.Elevation

object

ElevationContours

The object or objects this property applies to.

Elevation

Double; read-write

The elevation of the Contours collection.

Remarks

When the elevation is set, the <u>ElevationContours</u> collection will be regenerated with the contours for that elevation. All contours in the collection are of the same elevation.

Specifies the elevation override for the project database point group.

Signature

object.Elevation

object

PointGroup

The object or objects this property applies to.

Elevation

String; read-write

Remarks

If the Elevation property is set, then the <u>ElevationXDRef</u> property is cleared.

Returns the Elevation of the PVI.

Signiture

Object.Elevation

object

<u>PVI</u>

The object or objects this property applies to.

Elevation

Double; read-only

The Elevation for the PVI.

The Elevation for the TinPoint

Signature

object.Elevation

object

TinPoint

The object or objects this property applies to.

Elevation

Double; read-only
The elevation for the point.

ElevationContours Property

Returns the GenerateContours collection.

See Also | Example

Signature

object. Elevation Contours

object

SurfaceOutputs The object or objects this property

applies to.

ElevationContours

ElevationContours collection; read-only

The **ElevationContours** collection.

ElevationOverride Property

Specifies if the elevation for the project database point will be overridden from the Point Group.

See Also | Example

Signature

object.ElevationOverride

object

<u>PointGroup</u> The object or objects this property applies to.

ElevationOverride

Boolean; read-write

TRUE: The Cogo project point elevation will be set

by the Point Group.

FALSE: The Cogo project point elevation will not be

set by the Point Group.

Remarks

When you set this value to FALSE, the <u>Elevation</u> property will be cleared.

ElevationPrecision Property

Specifies the elevation precision for the drawing.

See Also | Example

Signature

object.ElevationPrecision

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

ElevationPrecision

Long; read-write

The number of decimals to the right of the decimal

point.

Remarks

The precision settings are used only for labeling and listing values, not actual computations. AutoCAD Land Desktop commands always calculate all numbers up to the highest internal precision.

ElevationXDRef Property

Specifies the elevation override for the project database point group.

See Also | Example

Signature

object. Elevation XDRef

object

PointGroup The object or objects this property

applies to.

ElevationXDRef

String; read-write

The name of the external database reference.

Remarks

The elevation will be taken from the external database reference (XDRef). If the ElevationXDRef property is set, then the <u>Elevation</u> property is cleared.

EndDirection Property

The ending direction of the curve.

See Also | Example

Signature

object.EndDirection

object

AlignCurve The object or objects this property

applies to.

EndDirection

Double; read-only

The end angle of the curve.

Remarks

The direction is in radians and measured counter-clockwise from the X axis (East).

EndEasting Property

The Easting coordinate for the end of the individual object entities.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 ParcelEntity, ParcelCurve, ParcelLine

The Easting coordinate for the end of the individual alignment entities.

Signature

object.EndEasting

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

EndEasting

Double; read-only

The Easting coordinate for the end point.

The Easting coordinate for the end of the individual parcel entities.

Signature

object.EndEasting

object

ParcelEntity, ParcelCurve, ParcelLine

The object or objects this property applies to.

EndEasting

Double; read-only

The Easting coordinate for the end point.

EndingStation Property

The object's ending station.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 Alignment

1 ProfileBlock (Civil Engineering Feature)

The EndingStation for individual entities in an alignment.

Signature

object.EndingStation

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

StartingStation

Double; read-only

The ending station for the object.

The EndingStation for the overall alignment.

Signature

object.EndingStation

object

Alignment

The object or objects this property applies to.

EndingStation

Double; read-only

The ending station for the object.

Returns the EndingStation for the Profile.

Signature

object.EndStation

object

ProfileBlock

The object or objects this property applies to.

EndStation

Double; read-only The Ending Station for the Profile.

EndNorthing Property

The Northing coordinate for the end of the individual object entity.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 ParcelEntity, ParcelCurve, ParcelLine

The Northing coordinate for the end of the individual alignment entities.

Signature

object.Insert

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

EndNorthiing

Double; read-only

The Northing coordinate for the end point.

The Northing coordinate for the end of the individual parcel entities.

Signature

object.Insert

object

ParcelEntity, ParcelCurve, ParcelLine

The object or objects this property applies to.

EndNorthiing

Double; read-only

The Northing coordinate for the end point.

ExceedBoth Property

If true, then both minimum depression settings must be met.

See Also | Example

Signature

object.ExceedBoth

object

WaterSheds The object or objects this property

applies to.

ExceedBoth

Boolean; read-only

TRUE: Both minimum depression conditions must

be met.

FALSE: Both minimum depression conditions do not

need to be met.

ExtEasting Property

The Easting coordinate for the External point (Ext).

See Also | Example

Signature

object.ExtEasting

object

AlignSpiral The object or objects this property

applies to.

ExtEasting

Double; read-only

The Easting coordinate for the External point.

Remarks

The External point is used for compound spirals, where the TS point is not on the spiral. This would mark the point where the spiral begins or ends.

ExternalSecant Property

The external secant length for the curve.

See Also | Example

Signature

object.ExternalSecant

object

AlignCurve The object or objects this property

applies to.

ExternalSecant

Double; read-only

The external secant length for the curve.

ExtNorthing Property

The Northing coordinate for the External point (Ext).

See Also | Example

Signature

object.ExtNorthing

object

AlignSpiral The object or objects this property

applies to.

ExtNorthing

Double; read-only

The Northing coordinate for the External point.

Remarks

The External point is used for compound spirals, where the TS point is not on the spiral. This would mark the point where the spiral begins or ends.

Faces Property

Returns the Faces collection.

See Also | Example

Signature

object.Faces

object

SurfaceOutputs The object or objects this property

applies to.

Faces

Faces collection; read-only

The <u>Faces</u> collection.

FacetDeviation Property

Gets and sets the facet deviation.

See Also | Example

Signature

object.FacetDeviation

object

DatabasePreferences The object or objects this

property applies to.

FacetDeviation

Double; read-write The facet deviation.

FGPrecision Property (Civil Engineering Feature)

Returns the finished ground precision for the alignment profile or cross section.

See Also | Example

Signature: Overview

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 <u>ProfileBlock</u> (Civil Engineering Feature)

Returns the finished ground precision for the alignment cross section.

Signature

object.FGPrecision

object

CrossSectionBlock The object or objects this

property applies to.

FGPrecision

Integer; read-only

The finished ground precision for the alignment

profile.

Returns the finished ground precision for the alignment profile.

Signature

object.FGPrecision

object

ProfileBlock

The object or objects this property applies to.

FGPrecision

Integer; read-only

The finished ground precision for the alignment

profile.

FGProfiles Property (Civil Engineering Feature)

The proposed vertical alignment's entities collection.

See Also | Example

Signature

object.ProposedVerticalAligns

object

Alignment The object or objects this property

applies to.

FGProfiles

FGProfiles collection; read-only

The **FGProfiles** collection.

File Property

The filename of the file lock.

See Also | Example

Signature

object.File

object

FileLock The object or objects this property applies

to.

File

String; read-only

The filename and path of the lock file.

FileLocks Property

Gets the FileLocks collection.

See Also | Example

Signature

object.FileLocks

object

Project The object or objects this property applies

to.

FileLocks

FileLocks collection; read-only

The FileLocks collection.

Remarks

The FileLocks collection represents all file locks for this project.

Files Property

Gets the PreferencesFiles object.

See Also | Example

Signature

object.Files

object

Preferences The object or objects this property

applies to.

Files

<u>PreferencesFiles</u> object; read-only Gets the <u>PreferencesFiles</u> object.

Remarks

This object specifies the paths used by the program. Some paths are established during installation and are exposed as read-only properties.

FirstTimeDrawingSetup Property

Selects how new drawings are setup.

See Also | Example

Signature

object.FirstTimeDrawingSetup

object

<u>PreferencesUser</u> The object or objects this property applies to.

First Time Drawing Setup

eAeccFirstTimeDrawingSetup enum; read-write The type of drawing setup.

kDrawingSetupWizard:

Use the Drawing Setup

Wizard

kDrawingSetupCommand:

Use the Drawing Setup

command

kAutoload Setup File:

Automatically load a

setup file

Remarks

The initial value for this property is kDrawingSetupWizard.

If you choose kAutoloadSetupFile, the file used is specified by the <u>FirstTimeDrawingSetupFile</u> property.

FirstTimeDrawingSetupFile Property

The drawing setup file loaded automatically for a new drawing.

See Also | Example

Signature

object. First Time Drawing Setup File

object

<u>PreferencesUser</u> The object or objects this property applies to.

First Time Drawing Setup File

String; read-write

The name of the setup file that will be loaded.

Format Property

The format for the PointFile.

See Also | Example

Signature

object.Format

object

PointFile The object or objects this property applies

to.

Format

String; read-only

The format for the input file.

Remarks

The input file must conform to the standard NEZ (space-delimited) file format.

FormatsPath Property

Specifies the directory in which the point import and export formats are located.

See Also | Example

Signature

object.FormatPath

object

PreferencesFiles The object or objects this property

applies to.

FormatPath

String; read-write

The drive and path for the point import and export

format files.

Remarks

By default, the project path is C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data\Format Manager. This path is not created during install. The commands that support point import and export formats creates this path when required. You must exit AutoCAD Land Desktop and restart for this change to take effect.

FullDescription Property

Returns the full description for the Cogo Point.

See Also | Example

Signature

object.FullDescription

object

CogoPoint The object or objects this property

applies to.

FullDescription

String; read-only

Remarks

The FullDescription includes DescriptionKey Overrides.

FullName Property

The full name includes the path and file name.

See Also | Example

Signature: Overview

- 1 DescriptionKeyFile
- 1 Drawing
- 1 Project

The full name for the DescriptionKeyFile for a project. The full name includes the path and file name.

Signature

object.FullName

object

DescriptionKeyFile;

The object or objects this property applies to.

FullName

String; read-only

Gets the name of the drawing, including the path.

Signature

object.FullName

object

Drawing

The object or objects this property applies to.

FullName

String; read-only

The name and path of the application or document.

Remarks

This value is read-only and cannot be changed.

Specifies the name and path of the project.

Signature

object.FullName

object

Project

The object or objects this property applies to.

FullName

String; read-only

The name and path of the project.

Remarks

This property returns the fully qualified path and name of the project.

GridEasting Property

The grid Easting for the COGO project database point.

See Also | Example

Signature

object.GridEasting

object

CogoPoint The object or objects this property

applies to.

GridEasting

Double; read-only

GridNorthing Property

The grid Northing for the COGO project database point.

See Also | Example

Signature

object.GridNorthing

object

CogoPoint The object or objects this property

applies to.

GridNorthing

Double; read-only

GroupName Property

Gets / sets the group name for various objects.

See Also | Example

Signature: Overview

l CogoPoint

1 PointGroup

Gets / sets the group name for the Point.

Signature

object.GroupName

object

CogoPoint The object or objects this property

applies to.

GroupName

String; read-write

Returns and sets the group name for the

CogoPoints collection.

Remarks

If no GroupName has been set, then the <u>OverrideDescription</u> and <u>OverrideName</u> will be empty. <u>OverrideElevation</u> will be -1E20. If

GroupName has been set, then these values will reflect the PointGroup overrides. If the PointGroup does not have an override for a field, the field will not change.

Gets / sets the group name for the PointGroup.

Signature

object.GroupName

object

PointGroup

The object or objects this property applies to.

GroupName

String; read-write

The group name for the PointGroup.

Height Property (Civil Engineering Feature)

Returns the height of the alignment cross section.

See Also | Example

Signature

object.Height

object

CrossSectionBlock The object or objects this

property applies to.

Height

Double; read-only

The height of the alignment cross section.

HelpPath Property

Specifies the directory in which the help files are installed.

See Also | Example

Signature

object.HelpPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

HelpPath

String; read-only

The drive and path for the help files.

Id Property

The identification number for the Boundary, BreakLine, ContourItem or Watershed.

See Also | Example

Signature: Overview

- l **Boundary**
- 1 BreakLine
- 1 ContourItem
- 1 WaterShed

An identification number for the Boundary.

Signature

object.Id

object

Boundary The object or objects this property applies

to.

Id

Long: read-only

The identification number of the boundary.

Remarks

The Boundaries class uses the Boundary <u>Id</u> to delete and find an instance of a boundary.

The identification number for the BreakLine.

Signature

object.Id

object

BreakLine

The object or objects this property applies to.

Id

Long; read-only

The identifier for the BreakLine.

An identification number for the ContourItem.

Signature

object.Id

object

ContourItem

The object or objects this property applies to.

Id

Long: read-only

The identification number of the contour.

Remarks

The ContourItems class uses the ContourItem <u>Id</u> to delete and find an instance of a contour.

An identification number for the WaterShed.

Signature

object.Id

object

WaterShed

The object or objects this property applies to.

Id

Long: read-only
The identification number of the WaterShed.

Inputs Property

Returns the SurfaceInputs object for the surface.

See Also | Example

Signature

object.Inputs

object

Surface The object or objects this property applies

to.

Inputs

<u>SurfaceInputs</u>; read-only All inputs for the surface.

IsBreakLine Property

Is the Boundary considered a BreakLine?

See Also | Example

Signature

object.IsBreakLine

object

Boundary The object or objects this property applies

to.

IsBreakLine

Boolean; read-write

TRUE: The Boundary will be considered as a

breakline

FALSE: The boundary will not be considered as a

breakline.

Remarks

If the Boundary is considered a breakline, faces will be generated along the Boundary.

IsNameSupported Property

Returns whether point names are supported or not.

See Also | Example

Signature

object.IsNameSupported

object

CogoPoints The object or objects this property

applies to.

IsNameSupported

Boolean; read-only

TRUE: Point names are supported. FALSE: Point names are not supported.

IsVisible Property

Returns if the Face is visible.

See Also | Example

Signature

object.IsVisible

object

Face The object or objects this property applies to.

IsVisible

Boolean; read-only

TRUE: If the Face is visible. FALSE: If the Face is not visible.

Remarks

Portions of the Surface can be made invisible by using boundaries. This does not mean that the Face does not exist in the Surface. It only means that the face is not shown when importing the Surface.

K Property

The spiral's K value.

See Also | Example

Signature

object.K

object

AlignSpiral The object or objects this property

applies to.

K

Double; read-only The spiral's K value.

Keywords Property

Specifies the keywords for the project.

See Also | Example

Signature

object.Keywords

object

Project The object or objects this property applies

to.

Keywords

String; read-write

The keywords for the project.

Label Property

The name of the locked datafile or folder.

See Also | Example

Signature

object.Label

object

FileLock The object or objects this property applies

to.

Label

String; read-only

The label of the file lock.

LabelPoints Property

Returns the contour label points.

See Also | Example

Signature

object.LabelPoints

object

AeccContour The object or objects this property

applies to.

LabelPoints

Variant (three-element array of doubles); read-only

The array of points for the contour labels.

LabelStyle Property

Specifies the label style override for the project database point group.

See Also | Example

Signature

object.LabelStyle

object

PointGroup The object or objects this property

applies to.

LabelStyle

String; read-write

The label style override for the project database

point group.

LabelStyleOverride Property

Specifies if the label style for the project database point will be overridden from the Point Group.

See Also | Example

Signature

object.LabelStyleOverride

object

<u>PointGroup</u> The object or objects this property applies to.

LabelStyleOverride

Boolean; read-write

TRUE: The point group label style will be

overridden.

FALSE: The point group label style will not be

overridden.

LabelStylePath Property

Specifies the directory in which the line, curve, spiral and point label style files are located.

See Also | Example

Signature

object.LabelStylePath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

Label Style Path

String; read-write

The drive and path for the label style files.

Remarks

By default, the project path is C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data\labels. You must exit AutoCAD Land Desktop and restart for this change to take effect.

LabelStyleXDRef Property

Specifies the label style override for the project database point group.

See Also | Example

Signature

object. Label Style XDRef

object

PointGroup The object or objects this property

applies to.

LabelStyleXDRef

String; read-write

The label style override for the project database

point group.

LastUsedDwg Property

The last used drawing.

See Also | Example

Signature

object.LastUsedDwg

object

<u>PreferencesUser</u> The object or objects this property

applies to.

LastUsedDwg

String; read-only

The name of the drawing that was last used.

LastUsedDwgPath Property

The last used drawing path.

See Also | Example

Signature

object. Last Used Dwg Path

object

<u>PreferencesUser</u> The object or objects this property

applies to.

LastUsedDwgPath

String; read-only

The name of the drawing path that was last used.

LastUsedProj Property

The last used project.

See Also | Example

Signature

object.LastUsedProj

object

<u>PreferencesUser</u> The object or objects this property

applies to.

Last Used Proj

String; read-only

The name of the project that was last used.

LastUsedProjPath Property

The last used project path.

See Also | Example

Signature

object. Last Used Proj Path

object

<u>PreferencesUser</u> The object or objects this property

applies to.

LastUsedProjPath

String; read-only

The name of the project path that was last used.

Latitude Property

The latitude of the point.

See Also | Example

Signature

object.Latitude

object

CogoPoint The object or objects this property

applies to.

Latitude

Double; read-only

LayerFile Property

The layer file to apply to the drawing.

See Also | Example

Signature

object.LayerFile

object

DatabasePreferences The object or objects this

property applies to.

LayerFile

String; read-write

The name of the layer file.

Remarks

The layer file is stored as a global setting in the registry, and not on a perdrawing basis. An entry is maintained for each <u>MeasurmentUnit</u>.

LayerStandard Property

The layer standard to apply to the drawing.

See Also | Example

Signature

object.LayerStandard

object

DatabasePreferences The object or objects this

property applies to.

LayerStandard

String; read-write

The name of the layer standard.

Remarks

The layer standards are stored on a per drawing basis in the dictionary.

LeftWidth Property (Civil Engineering Feature)

Returns the left width of the alignment cross section.

See Also | Example

Signature

object.LeftWidth

object

CrossSectionBlock The object or objects this

property applies to.

LeftWidth

Double; read-only

The left width of the alignment cross section.

Length Property

The length of the alignment curve, spiral or tangent or parcel curve or line.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 ParcelEntity, ParcelCurve, ParcelLine

The length of the alignment curve, spiral or tangent.

Signature

object.Length

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

Length

Double; read-only

The length of the alignment curve, spiral or line.

The length of the parcel curve or line.

Signature

object.Length

object

ParcelEntity, ParcelCurve, ParcelLine
The object or objects this property applies to.

Length

Double; read-only

The length of the parcel curve or line.

LExt Property

Returns the external length of the spiral.

See Also | Example

Signature

object.LExt

object

AlignSpiral The object or objects this property

applies to.

LExt

Double; read-only

The external length of the spiral.

Remarks

The LExt is the external length of the spiral. For a simple spiral, the value is 0.0. This represents the length from the TS to the External point in a compound spiral condition.

LinearDisplayFormat Property

The format used to display linear values.

See Also | Example

Signature

object.LinearDisplayFormat

object

<u>DatabasePreferences</u> The object or objects this property applies to.

LinearDisplayFormat

Long; read-write

- 1: Scientific
- 2: Decimal
- 3: Engineering
- 4: Architectural
- 5: Fractional

LinearPrecision Property

Specifies the linear precision for the drawing.

See Also | Example

Signature

object.LinearPrecision

object

DatabasePreferences The object or objects this

property applies to.

LinearPrecision

Long; read-write

The number of decimals to the right of the decimal

point.

Remarks

The precision settings are used only for labeling and listing values, not actual computations. AutoCADLand Desktop commands always calculate all numbers up to the highest internal precision.

LinearUnit Property

The unit used to display linear values.

See Also | Example

Signature

object.LinearUnit

object

DatabasePreferences The object or objects this

property applies to.

LinearUnits

AecBuiltInUnit enum; read-write

aecUnitInch: Inch units

aecUnitFoot: Foot units

aecUnitMillimeter: Millimeter units

aecUnitCentimeter: Centimeter units

aecUnitDecimeter: Decimeter units

aecUnitMeter: Meter units

Remarks

Use this property to specify whether you want to use feet or meters in your drawing.

The linear units that you select must be the same for all drawings in a project. You cannot mix foot-based drawings with meter-based drawings in the same project.

LockedPointNumbers Property

Returns a list of the locked points in the current project point database.

See Also | Example

Signature

object.LockedPointNumbers()

object

<u>CogoPoints</u> The object or objects this property applies to.

LockedPointNumbers

String; read-only

Comma delimited string with all used points with groups separated by hyphens.

Remarks

Typical examples of the point string format are: "1,2,3,5,10" "1-100,1000-2000"

LockType Property

Specifies the type of lock on the Alignment, COGO Point, or Surface.

See Also | Example

Signature

object.LockType

object

Alignment, CogoPoint, FileLock, Surface

The object or objects this property applies to.

LockType

eAeccFileLockType; read-only

kNoLock: No lock exists

kReadLock: User can only read the object

kWriteLock: User can read and write the object

LOffset Property

The offset distance for the spiral.

See Also | Example

Signature

object.LOffset

object

AlignSpiral The object or objects this property

applies to.

LOffset

Double; read-only

The offset distance for the spiral.

Remarks

If the spiral is an offset spiral (see property SpiralType2), this property is set. Otherwise, it is 0.0.

Longitude Property

The longitude of a point.

See Also | Example

Signature

object.Longitude

object

CogoPoint The object or objects this property

applies to.

Longitude

Double; read-only

MaxElevation Property

Returns the Maximum Elevation of a Cross Section or Surface.

See Also | Example

Signature: Overview

1 <u>CrossSection</u> (Civil Engineering Feature)

1 Surface

Returns the Maximum Elevation of a Cross Section.

Signature

RetVal = object.MaxElevation

object

CrossSection The object or objects this property

applies to.

MaxElevation

Double; read-only

The Maximum Elevation of the Cross Section.

Returns the Maximum Elevation of a Surface.

Signature

RetVal = object.MaxElevation

object

Surface

The object or objects this property applies to.

MaxElevation

Double; read-only
The Maximum Elevation of the Surface.

MaxFaceArea Property

The area of the largest Face for the Surface.

See Also | Example

Signature

object.MaxFaceArea

object

Surface The object or objects this property applies

to.

MaxFaceArea

Double; read-only

The area of the largest Face.

Remarks

This is an Extended Statistic.

MaxGrade Property

The maximum grade for the Surface.

See Also | Example

Signature

object.MaxGrade

object

Surface The object or objects this property applies

to.

MaxGrade

Double; read-only

The maximum grade for the surface.

Remarks

This is an Extended Statistic.

MaxOffset Property (Civil Engineering Feature)

Returns the Maximum Offset of the Cross Section.

See Also | Example

Signature

RetVal = object.MaxOffset

object

CrossSection The object or objects this property

applies to.

MaxOffset

Double; read-only

The Maximum Offset of the Cross Section.

MeanElevation Property

The mean elevation of the Surface.

See Also | Example

Signature

object.MeanElevation

object

Surface The object or objects this property applies

to.

MeanElevation

Double; read-only

The mean elevation of the Surface.

Remarks

This is an Extended Statistic.

MeasurementUnit Property

The system of units used to display measurements.

See Also | Example

Signature

object.MeasurementUnit

object

DatabasePreferences The object or objects this

property applies to.

MeasurementUnit

AcMeasurementUnit enum; read-only

acEnglish: English units

acMetric: Metric units

MidOrdinate Property

The mid ordinate for the curve.

See Also | Example

Signature

object.MidOrdinate

object

AlignCurve The object or objects this property

applies to.

MidOrdinate

Double; read-only

The mid ordinate length for the curve.

MinDepressionArea Property

The minimum depression area for all watersheds.

See Also | Example

Signature

object. M in Depression Area

object

WaterSheds The object or objects this property applies to.

MinDepression Area

Double; read-only

The minimum depression area for all watersheds.

MinDepressionDepth Property

The minimum depression depth for all watersheds.

See Also | Example

Signature

object. M in Depression Depth

object

WaterSheds The object or objects this property applies to.

MinDepressionDepth

Double; read-only

The minimum depression depth for all watersheds.

MinElevation Property

Returns a Minimum Elevation of a Cross Section or Surface.

See Also | Example

Signature: Overview

1 <u>CrossSection</u> (Civil Engineering Feature)

1 Surface

Returns the Minimum Elevation of a Cross Section.

Signature

RetVal = object.MinElevation

object

CrossSection The object or objects this property

applies to.

MinElevation

Double; read-only

The Minimum Elevation of the Cross Section.

Returns the Minimum Elevation of a Surface.

Signature

RetVal = object.MinElevation

object

Surface

The object or objects this property applies to.

MinElevation

Double; read-only The Minimum Elevation of the Surface.

MinFaceArea Property

The area of the smallest Face for the Surface.

See Also | Example

Signature

object.MinFaceArea

object

Surface The object or objects this property applies

to.

MinFaceArea

Double; read-only

The area of the smallest Face.

Remarks

This is an Extended Statistic.

MinGrade Property

The minimum grade for the Surface.

See Also | Example

Signature

object.MinGrade

object

Surface The object or objects this property applies

to.

MinGrade

Double; read-only

The minimum grade for the surface.

Remarks

This is an Extended Statistic.

MinOffset Property (Civil Engineering Feature)

Returns the Minimum Offset of the Cross Section.

See Also | Example

Signature

RetVal = object.MinOffset

object

CrossSection The object or objects this property

applies to.

MinOffset

Double; read-only

The Minimum Offset of the Cross Section.

Name Property

Gets the name of various objects.

See Also | Example

Signature: Overview

- l AeccContourStyle
- 1 Alignment
- l CogoPoint
- 1 <u>CrossSectionBlock</u> (Civil Engineering Feature)
- l <u>CrossSectionSurface</u> (Civil Engineering Feature)
- l **DEMFile**
- l <u>DescriptionKeyFile</u>
- l <u>Drawing</u>
- l <u>FGProfiles</u> (Civil Engineering Feature)
- 1 Parcel
- l <u>PointFile</u>
- l PointGroup
- 1 PointGroupName

1 ProfileBlock (Civil Engineering Feature) 1 Project 1 Prototype 1 Surface The name of the contour style. **Signature** object.Name object AeccContourStyle The object or objects this property applies to. Name String; read-only The name of the contour style. The name of the alignment. **Signature** object.Name object **Alignment** The object or objects this property applies to. Name String; read-only The name of the alignment.

The point name.

Signature

object.Name

object

CogoPoint

The object or objects this property applies to.

Name

String; read-only

Remarks

When the project is created, an option exists to create Cogo points with Names. If the project does not use point names, then this property will be empty and can not be set.

Returns the name of the cross sections alignment profile.

Signature

object.Name

object

CrossSectionBlock

The object or objects this property applies to.

Name

String; read-only

The name of the cross sections alignment profile.

The cross section surface name.

Signature

object.Name

object

CrossSectionSurface

The object or objects this property applies to.

Name

String; read-only

The cross section surface name.

The name of the Digital Elevation Model file to be included in the Surface definition.

Signature

object.Name

object

DEMFile;

The object or objects this property applies to.

Name

String; read-only

The fully qualified name and path of the input file.

The Name for the description key file.

Signature

object.Name

object

DescriptionKeyFile;

The object or objects this property applies to.

Name

String; read-write

The name for the description key file, not including

the path.

Gets the name of the drawing, not including the path.

Signature

object.Name

object

Drawing

The object or objects this property applies to.

Name

String; read-only

The name of the drawing.

Returns the Finished Grade Profiles alignment name.

Signiture

Object.Name

object

FGProfiles

The object or objects this property applies to.

Name

String; read-only

The name of the Finished Grade Profile alignment.

The name of the parcel.

Signature

object.Name

object

Parcel

The object or objects this property applies to.

Name

String; read-only

The name of the parcel.

The name of the file to be included in the Surface definition.

Signature

object.Name

object

PointFile

The object or objects this property applies to.

Name

String; read-only

The name of the input file.

Specifies the Name override for the project database point group.

Signature

object.Name

object

PointGroup

The object or objects this property applies to.

Name

String; read-write

Remarks

If the Name property is set, then the <u>NameXDRef</u> property is cleared.

The Name of the PointGroup to be used to build the Surface.

Signature

object.Name

object

PointGroupName

The object or objects this property applies to.

Name

String; read-only

Name of the PointGroup.

Remarks

The PointGroupName is a link between Cogo Point Groups and TM. By using a PointGroupName, that Cogo Point Group will be included in the Surface definition.

No validation exists between a PointGroupName and a Cogo Point Group. If the PointGroupName.Name does not exist as a Cogo Point Group, no errors will be generated.

Returns the Alignment name for the Profile.

Signiture

Object.Name

object

ProfileBlock

The object or objects this property applies to.

Name

String; read-write

The Alignment name for the Profile.

Specifies the name of the project.

Signature

object.Name

object

Project

The object or objects this property applies to.

Name

String; read-only

The name of the project.

Remarks

This property returns the directory name only, not the Path.

The name of the prototype.

Signature

object.Name

object

Prototype

The object or objects this property applies to.

Name

String; read-write

The name of the prototype.

Remarks

Use this property to rename a prototype.

The Name of the Surface.

Signature

object.Name

object

Surface

The object or objects this property applies to.

Name

String; read-only
The Name of the Surface.

Remarks

Surface names are limited to 40 characters.

NameOverride Property

Specifies if the name for the project database point will be overridden from the Point Group.

See Also | Example

Signature

object.NameOverride

object

PointGroup The object or objects this property

applies to.

NameOverride

Boolean; read-write

TRUE: The Cogo project point name will be set by

the Point Group.

FALSE: The Cogo project point name will not be set

by the Point Group.

Remarks

When you set this value to FALSE, the <u>Name</u> property will be cleared.

NameXDRef Property

Specifies the Name override for the project database point group. The name will be taken from the external database reference (XDRef).

See Also | Example

Signature

object.NameXDRef

object

PointGroup The object or objects this property

applies to.

NameXDRef

String; read-write

The name of the external database reference.

Remarks

If the NameXDRef property is set, then the Name property is cleared.

NextPointNumber Property

Gets / sets the next sequential point number for point creation.

See Also | Example

Signature

object.NextPointNumber

object

CogoPoints The object or objects this property

applies to.

NextPointNumber

Long; read-write

Remarks

The default number when you begin a project is 1. An error is returned if you try to set the property to a value less than 1 or a used point number.

Normal Property

Returns the Normal for the Face.

See Also | Example

Signature

object.Normal

object

Face The object or objects this property applies to.

Normal

Variant (3 element array of doubles); read-only

Returns the normal for the Face.

Northing Property

Gets the Northing coordinates of various objects.

See Also | Example

Signature: Overview

1 AeccPoint

l CogoPoint

1 TinPoint

The Northing coordinate of the point.

Signature

object.Northing

object

AeccPoint The object or objects this property

applies to.

Northing

Double; read-write

The Northing coordinate of the point.

Remarks

The Easting coordinate is derived from the WCS X coordinate with the <u>BasePoint</u> and <u>NorthRotation</u> applied.

The north coordinate for the point.

Signature

object.Northing

object

CogoPoint

The object or objects this property applies to.

Northing

Double; read-write

The Northing coordinate for the TinPoint.

Signature

object.Northing

object

TinPoint

The object or objects this property applies to.

Northing

Double; read-only

The Northing coordinate.

NorthRotation Property

Specifies the north rotation for your drawing layout.

See Also | Example

Signature

object.NorthRotation

object

DatabasePreferences The object or objects this

property applies to.

NorthRotation

Double; read-write

The north rotation angle. The angle is in radians and measured counter-clockwise from the X axis (East).

Remarks

The initial value for this property is 0.

If you change the NorthRotation in an existing drawing then you must move any objects in the drawing to match the new coordinate system.

Number Property

The number property for various objects.

See Also | Example

Signature: Overview

l AeccPoint

l <u>Alignment</u>

l CogoPoint

1 Parcel

The AeccPoint point number.

Signature

object.Number

object

AeccPoint The object or objects this property

applies to.

Number

Long; read-only The point number.

Remarks

The number property must be unique if the AeccPoint is to be imported into the Cogo database.

The Alignment number.

Signature

object.Number

object

Alignment

The object or objects this property applies to.

Number

Long; read-only

The number of the alignment.

The CogoPoint number.

Signature

object.Number

object

CogoPoint

The object or objects this property applies to.

Number

Long; read-only

Remarks

The point number is set with the value of the <u>NextPointNumber</u> property of the <u>CogoPoints</u> collection when the <u>Add</u> method is invoked, and cannot be changed.

The number of the Parcel.

Signature

object.Number

object

<u>Parcel</u>

The object or objects this property applies to.

Number

Long; read-only
The number of the parcel.

NumberOfFaces Property

The number of Faces for the Surface.

See Also | Example

Signature

object.NumberOfFaces

object

Surface The object or objects this property applies

to.

NumberOfFaces

Long; read-only

The number of faces in the surface.

Remarks

This is an Extended Statistic.

NumberOfPoints Property

The number of TinPoints for the Surface.

See Also | Example

Signature

object. Number Of Points

object

Surface The object or objects this property applies

to.

NumberOfPoints

Long; read-only

The number of points in the surface.

ObjectID Property

Returns the Object ID for the entity.

See Also | Example

Signature

object.ObjectID

object

AlignCurve, AlignEntity, AlignSpiral, AlignTangent

The object this property applies to.

ObjectID

Long; read-only

The Object ID for the entity.

Remarks

An Alignment can exist in multiple drawings. The Object ID for the entity is for the current drawing only.

Valid alignment ObjectIDs are defined during an <u>Import</u>. The ObjectIDs are valid until the alignment is saved. Alignments are automatically saved each time an alignment property is modified when <u>AutoSave</u> is set to true, or when an alignment <u>Save</u> is executed.

Offset Property (Civil Engineering Feature)

Returns the cross section point code offset.

See Also | Example

Signature

object.Offset

object

CrossSectionPointCode The object or objects this

property applies to.

Offset

Double; read-only

The cross section point code offset.

OffsetElevations Property (Civil Engineering Feature)

Returns the cross section surface offset and elevations.

See Also | Example

Signature

object.OffsetElevations

object

CrossSectionSurface The object or objects this

property applies to.

OffsetElevations

Variant (array of doubles); read-only

Returns the cross section surface offset and

elevations.

Outputs Property

Returns the SurfaceOutputs object for the surface.

See Also | Example

Signature

object.Outputs

object

Surface The object or objects this property applies

to.

Inputs

<u>SurfaceOutputs</u>; read-only All outputs for the surface.

OverflowPoints Property

Returns the WaterShed overflow points.

See Also | Example

Signature

object.OverflowPoints

object

WaterShed The object or objects this property

applies to.

OverflowPoints

Variant (3-element array of doubles); read-only The points that make up the calculated watershed.

OverrideDescription Property

Returns the description that has been overridden by the PointGroup.

See Also | Example

Signature

object.OverrideDescription

object

CogoPoint The object or objects this property applies to.

OverrideDescription

String; read-only

Remarks

If the <u>GroupName</u> has been set for the <u>CogoPoints</u> collection and that <u>PointGroup</u> has the description override, this value will be set.

OverrideElevation Property

Returns the elevation that has been overridden by the PointGroup.

See Also | Example

Signature

object. Override Elevation

object

CogoPoint The object or objects this property

applies to.

OverrideElevation

Double; read-only

Remarks

If the <u>GroupName</u> has been set for the <u>CogoPoints</u> collection and that <u>PointGroup</u> has the elevation override, this value will be set.

OverrideName Property

Returns the point name that has been overridden by the PointGroup.

See Also | Example

Signature

object.OverrideName

object

CogoPoint The object or objects this property

applies to.

OverrideName

String; read-only

Remarks

If the <u>GroupName</u> has been set for the <u>CogoPoints</u> collection and that <u>PointGroup</u> has the name override, this value will be set.

OverrideNew Property

Toggles the display of the AutoCAD Land Desktop New dialog box.

See Also | Example

Signature

object.OverrideNew

object

PreferencesUser The object or objects this property

applies to.

OverrideNew

Boolean; read-write

TRUE: Displays the AutoCAD Land Desktop New

dialog box.

FALSE: Displays the AutoCAD New dialog box.

Remarks

The initial value for this property is TRUE.

If you set this property to FALSE and use the AutoCAD New command to create a new drawing, then you must save the new drawing to name it. Only named drawings can be associated with a project.

You must exit AutoCAD Land Desktop and restart for this change to take effect.

OverrideOpen Property

Toggles the display of the AutoCAD Land Desktop Open dialog box.

See Also | Example

Signature

object.OverrideOpen

object

<u>PreferencesUser</u> The object or objects this property

applies to.

OverrideOpen

Boolean; read-write

TRUE: Displays the AutoCAD Land Desktop Open

dialog box.

FALSE: Displays the AutoCAD open dialog box.

Remarks

The initial value for this property is TRUE.

If you set this property to FALSE (and use the AutoCAD Open command to open a drawing), then you will automatically be prompted to select a project if the drawing isn't associated with a project or if the drawing's project is not found.

You must exit AutoCAD Land Desktop and restart for this change to take effect.

Owner Property

The owner (AutoCAD login name) of the lock owner.

See Also | Example

Signature

object.Owner

object

FileLock The object or objects this property applies

to.

Owner

String; read-only

The owner of the file lock.

P Property

The spiral's P value.

See Also | Example

Signature

object.P

object

AlignSpiral The object or objects this property

applies to.

P

Double; read-only The spiral's P value.

Parcel Property

The collection of parcel preferences for the project.

See Also | Example

Signature

object.Parcel

object

PreferencesProject The object or objects this

property applies to.

Parcel

<u>PreferencesParcel</u>; read-only The <u>PreferencesParcel</u> object.

ParcelEntities Property

The collection of geometric entities for the parcel.

See Also | Example

Signature

object.ParcelEntities

object

Parcel The object or objects this property applies to.

ParcelEntities

ParcelEntities collection; read-only

The <u>ParcelEntities</u> collection.

Parcels Property

Gets the Parcels collection.

See Also | Example

Signature

object.Parcels

object

Project The object or objects this property applies

to.

Parcels

Parcels collection; read-only

The Parcels collection.

Remarks

The <u>Parcels</u> collection represents all of the <u>Parcel</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

Path Property

Gets the path of various objects.

See Also | Example

Signature: Overview

1 <u>DescriptionKeyFile</u>

1 DescriptionKeyFiles

1 Drawing

l **Drawings**

1 FileLocks

l Project

The path for the description key files for a project.

Signature

object.Path

object

DescriptionKeyFile, DescriptionKeyFiles

The object or objects this property applies to.

Path

String; read-only

Gets the path of the drawing.

Signature

object.Path

object

Drawing

The object or objects this property applies to.

Path

String; read-only

The path of the drawing.

Remarks

This value is read-only and cannot be changed.

Returns the path at which the drawings for this project are located.

Signature

object.Path

object

Drawings

The object or objects this property applies to.

Path

String; read-only

The keyword for the project.

Specifies the path at which the project locks are located.

Signature

object.Path

object

FileLocks

The object or objects this property applies to.

Path

String; read-only

The path name to for the collection.

Remarks

Use this property to find locks for a given project.

Specifies the path of the project.

Signature

object.Path

object

Project

The object or objects this property applies to.

Path

String; read-only

The path of the project.

Remarks

The path of the project contains the <u>Name</u> as the last folder in the path specification.

Perimeter Property

The perimeter of the parcel.

See Also | Example

Signature

object.Perimeter

object

Parcel The object or objects this property applies to.

Perimeter

Double; read-only

The perimeter of the parcel.

PiEasting Property

The Easting coordinate for the curve's PI.

See Also | Example

Signature

object.PiEasting

object

AlignCurve The object or objects this property

applies to.

PiEasting

Double; read-only

The east coordinate for the PI of the curve.

PiNorthing Property

The Northing coordinate for the curve's PI.

See Also | Example

Signature

object.PiNorthing

object

AlignCurve The object or objects this property

applies to.

PiNorthing

Double; read-only

The north coordinate for the PI of the curve.

PointFiles Property

Returns the PointFiles collection.

See Also | Example

Signature

object.PointFiles

object

SurfaceInputs The object or objects this property

applies to.

PointFiles

PointFiles collection; read-only

The **PointFiles** collection.

PointGroupNames Property

Returns the PointGroupNames collection.

See Also | Example

Signature

object. Point Group Names

object

<u>SurfaceInputs</u> The object or objects this property

applies to.

PointGroupNames

PointGroupNames collection; read-only

The **PointGroupNames** collection.

PointGroups Property

Gets the PointGroups collection.

See Also | Example

Signature

object.PoinGroups

object

Project The object or objects this property applies

to.

PointGroups

PointGroups collection; read-only

The **PointGroups** collection.

Remarks

The <u>PointGroups</u> collection represents all of the <u>PointGroup</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

PointList Property

Specifies the list of project database points for the point group.

See Also | Example

Signature

object.PointList

object

PointGroup The object or objects this property

applies to.

PointList

String; read-write

Comma delimited string with all used points with

groups separated by hyphens.

Remarks

Examples: "1,2,3,5,10" "1-100,1000-2000"

PointNameSize Property

Returns maximum number of characters supported in point name.

See Also | Example

Signature

object.PointNameSize

object

CogoPoints The object or objects this property

applies to.

NextPointNumber

Long; read-only

Returns a long that represents the maimum number

of characters that a point name can hold.

PointOnLineTolerance Property

Returns the minimum distance between a line and a point not on the line.

See Also | Example

Signature

object. Point On Line Tolerance

object

<u>Surface</u> The object or objects this property applies to.

PointOnLineTolerance

Double; read-only

The point on line tolerance for the surface.

PointTolerance Property

Returns the minimum distance between distinct points.

See Also | Example

Signature

object.PointTolerance

object

Surface The object or objects this property applies

to.

PointTolerance

Double; read-only

The point tolerance for the surface.

Precision Property

The precision used to truncate area labels when a Parcel is imported to the drawing.

See Also | Example

Signature

object.Precision

object

Parcel The object or objects this property applies to.

Precision

Double; read-only

The parcel's Precision value.

Preferences Property

The preferences property of various objects.

See Also | Example

Signature: Overview

1 AeccApplication

1 Document

1 Project

Get the AeccApplication Preferences

Signature

object.Preferences

object

AeccApplication The object or objects this property

applies to.

Preferences

Preferences object; read-only

Remarks

The Preferences object holds the system level options.

Get the Preferences object.

Signature

object.Preferences

object

Document

The object or objects this property applies to.

Preferences

DatabasePreferences object; read-only

The <u>DatabasePreferences</u> object.

Remarks

The <u>DatabasePreferences</u> object holds the settings that are saved in the AutoCAD Land Desktop drawing.

Get the PreferencesProject object.

Signature

object.Preferences

object

Project

The object or objects this property applies to.

Preferences

PreferencesProject object; read-only

The <u>PreferencesProject</u> object.

Remarks

The Preferences object holds the project level options. This property is valid only for the <u>ActiveProject</u>.

PreferencesPath Property

Specifies the directory in which the preference settings are located.

See Also | Example

Signature

object.PreferencePath

object

PreferencesFiles The object or objects this property

applies to.

PreferencePath

String; read-write

The drive and path for the preferences.

Remarks

The preference settings are stored in the C:\Documents and Settings\All Users\Application Data\Autodesk\AutoCAD Land Desktop <version number>\<release number>\Data\pref folder. The file name is <AutoCAD login name>.dfm. The preference path settings are stored in the sdsk.dfm file in the program folder.

You must exit AutoCAD Land Desktop and restart for this change to take effect.

Profile Property (Civil Engineering Feature)

The collection of vertical profile preferences for the project.

See Also | Example

Signature

object.Profile

object

PreferencesProject The object or objects this

property applies to.

Profile

<u>PreferencesProfile</u>; read-only The <u>PreferencesProfile</u> object.

ProfileBlocks Property

Gets the ProfileBlocks object.

See Also | Example

Signature

object.ProfileBlock

object

Document The object or objects this property

applies to.

ProfileBlocks

ProfileBlocks object; read-only

The **ProfileBlocks** object.

ProgramPath Property

Specifies the directory in which the AutoCAD Land Desktop is installed.

See Also | Example

Signature

object.ProgramPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

ProgramPath

String; read-only

The drive and path where the AutoCAD Land

Desktop is installed.

ProjectName Property

The name of the project that the drawing belongs to.

See Also | Example

Signature

object.ProjectName

object

DatabasePreferences The object or objects this

property applies to.

ProjectName

String; read-write The project name.

Remarks

The value must 255 characters or less in length.

ProjectPath Property

Gets the project path of various objects.

See Also | Example

Signature: Overview

1 PreferencesFiles

l Projects

Specifies the directory in which the projects are located.

Signature

object.ProjectPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

ProjectPath

String; read-write

The drive and path for projects.

Remarks

By default, the project path is c:\Land Projects. You must exit AutoCAD Land Desktop and restart for this change to take effect.

Specifies the directory in which the projects are located.

Signature

object.ProjectPath

object

Projects

The object or objects this property applies to.

ProjectPath

String; read-write The project path.

Remarks

The initial value of this property is the same as the ProjectPath property of the <u>PreferencesFiles</u> object.

The members of this collection are AutoCAD Land Desktop projects located at this path.

Projects Property

Gets the Projects collection.

See Also | Example

Signature

object.Properties

object

AeccApplication The object or objects this property

applies to.

Projects

Projects collection; read-only

Remarks

The <u>Projects</u> collection allows you to access all the AutoCAD Land Desktop projects available on the network. This includes all project data stored outside the drawings.

PrototypeName Property

Specifies the name of the prototype used for the project.

See Also | Example

Signature

object.PrototypeName

object

Project The object or objects this property applies

to.

PrototypeName

String; read-only

The name of the prototype used for this project.

PrototypePath Property

Specifies the directory in which the project prototype files are located.

See Also | Example

Signature

object.PrototypePath

object

PreferencesFiles The object or objects this property

applies to.

PrototypePath

String; read-write

The drive and path for the project prototype files.

Remarks

By default, this path is C:\Documents and Settings\All Users\Application Data\AutoCAD Land Desktop <version number>\<release number>\Data\Prototypes. You must exit AutoCAD Land Desktop and restart for this change to take effect.

Prototypes Property

Gets the Prototypes collection.

See Also | Example

Signature

object.Prototypes

object

AeccApplication The object or objects this property

applies to.

Prototypes

Prototypes collection; read-only

Remarks

The <u>Prototypes</u> collection allows you to access all the prototypes available for configuring your next AutoCAD Land Desktop project.

PVIs Property (Civil Engineering Feature)

Gets the PVIs collection.

See Also | Example

Signature

object.PVIs

object

FGProfile The object or objects this property applies

to.

PVIs

PVIs collection; read-only

The PVIs collection.

RadialDistance Property

The radial distance of the spiral.

See Also | Example

Signature

object.RadialDistance

object

AlignSpiral The object or objects this property

applies to.

RadialDistance

Double; read-only

The radial distance at the SC.

Radius Property

The radius of the alignment or parcel curve.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The radius of the alignment curve.

Signature

object.Radius

object

AlignCurve The object or objects this property

applies to.

Radius

Double; read-only

The radius of the alignment curve.

The radius of the parcel curve.

Signature

object.Radius

object

ParcelCurve

The object or objects this property applies to.

Radius

Double; read-only
The radius of the parcel curve.

RawDescription Property

The raw description for the COGO project database point.

See Also | Example

Signature

object.RawDescription

object

CogoPoint The object or objects this property

applies to.

RawDescription

String; read-write

Remarks

The RawDescription for the Cogo point does not include PointGroup Overrides or DescriptionKeyFile. If a Description Key is used, then it is based on the RawDescription.

RevisionNumber Property

The revision number for the Surface.

See Also | Example

Signature

object.RevisionNumber

object

Surface The object or objects this property applies

to.

RevisionNumber

Long; read-only

The surface revision number.

Remarks

Each time the Surface is built, the revision number is updated.

RightWidth Property (Civil Engineering Feature)

Returns the right width of the alignment cross section.

See Also | Example

Signature

object.RightWidth

object

CrossSectionBlock The object or objects this

property applies to.

RightWidth

Double; read-only

The right width of the alignment cross section.

RotateByDescriptionParam Property

If TRUE, only the RotateDescriptionParam is rotated.

See Also | Example

Signature

object.RotateByDescriptionParam

object

<u>DescriptionKey</u> The object or objects this property applies to.

RotateByDescriptionParam

Boolean; read-write

TRUE: Only the <u>RotateDescriptionParam</u> is rotated. FALSE: Do not rotate by <u>RotateDescriptionParam</u> only.

RotateByFixedFactor Property

If TRUE, rotate by the RotateFactor.

See Also | Example

Signature

object.RotateByFixedFactor

object

<u>DescriptionKey</u> The object or objects this property applies to.

RotateByFixedFactor

Boolean; read-write

TRUE: Rotate by the RotateFixedFactor.

FALSE: Do not rotate by <u>RotateFixedFactor</u> only.

RotateClockwise Property

The rotation direction.

See Also | Example

Signature

object. Rotate Clockwise

object

<u>DescriptionKey</u> The object or objects this property

applies to.

RotateClockwise

Double; read-write The rotate direction.

RotateDescriptionParam Property

The description parameter for rotation.

See Also | Example

Signature

object. Rotate Description Param

object

<u>DescriptionKey</u> The object or objects this property applies to.

Rotate Description Param

Integer; read-write

Returns and sets the rotate description parameter.

RotateFixedFactor Property

The rotation factor.

See Also | Example

Signature

object. Rotate Fixed Factor

object

<u>DescriptionKey</u> The object or objects this property

applies to.

RotateFixedFactor

Double; read-write

The rotate fixed factor.

ScaleByDescriptionParam Property

If TRUE, only the ScaleDescriptionParam is scaled.

See Also | Example

Signature

object.ScaleByDescriptionParam

object

<u>DescriptionKey</u> The object or objects this property applies to.

ScaleByDescriptionParam

Boolean; read-write

TRUE: Only the <u>ScaleDescriptionParam</u> is scaled. FALSE: Not scaled by the <u>ScaleDescriptionParam</u>

ScaleByDrawingScale Property

If TRUE, the drawing scale factor is applied.

See Also | Example

Signature

object.ScaleByDrawingScale

object

<u>DescriptionKey</u> The object or objects this property applies to.

ScaleByDrawingScale

Boolean; read-write

TRUE: Scale by the drawing scale factor. FALSE: Do not scale by drawing scale factor.

ScaleByFixedFactor Property

If TRUE, the fixed scale factor <u>ScaleFactor</u> is applied.

See Also | Example

Signature

object.ScaleByFixedFactor

object

<u>DescriptionKey</u> The object or objects this property applies to.

ScaleByFixedFactor

Boolean; read-write

TRUE: Scale by the fixed scale factor

ScaleFixedFactor.

FALSE: Do not scale by fixed scale factor

ScaleFixedFactor.

ScaleDescriptionParam Property

The description parameter for scaling.

See Also | Example

Signature

object. Scale Description Param

object

<u>DescriptionKey</u> The object or objects this property applies to.

ScaleDescriptionParam

Integer; read-write

Returns and sets the scale description parameter.

ScaleFixedFactor Property

The fixed scale factor.

See Also | Example

Signature

object. Scale Fixed Factor

object

DescriptionKey The object or objects this property

applies to.

ScaleFixedFactor

Double; read-write

ScaleInXY Property

If TRUE, the symbol is scaled in the XY plane

See Also | Example

Signature

object.ScaleXY

object

DescriptionKey The object or objects this property

applies to.

ScaleXY

Boolean; read-write

TRUE: Symbol is scaled in the XY plane. FALSE: Do not scale in the XY plane.

ScaleInZ Property

If TRUE, the symbol is scaled in the Z axis.

See Also | Example

Signature

object.ScaleInZ

object

DescriptionKey The object or objects this property

applies to.

ScaleInZ

Boolean; read-write

TRUE: Symbol is scaled in the Z axis. FALSE: Do not scale in the Z axis.

ScaleOnInsert Property

Specifies whether objects are automatically scaled when inserted into the drawing.

See Also | Example

Signature

object.ScaleOnInsert

object

DatabasePreferences The object or objects this

property applies to.

ScaleOnInsert

Boolean; read-write

TRUE: New objects are scaled automatically.

FALSE: New objects are not scaled automatically.

SearchType Property

Returns how the Faces collection was generated.

See Also | Example

Signature

object.SearchType

object

Faces The object or objects this property applies to.

SearchType

eAeccSurfaceSearchType enum; read-only

kNoSearch: No filter has been applied

kSearchByPath: Search by path filter

kSearchByBoundary: Search by surface boundary

kSearchByPoint: Search by point filter

.

.

.

Remarks

The SearchType property describes how the <u>Faces</u> collection was generated. The filter returns only the faces that exist between two points.

SheetHeight Property

Specifies the height of the sheet.

See Also | Example

Signature

object.SheetHeight

object

DatabasePreferences The object or objects this

property applies to.

SheetHeight

Double; read-write The sheet height.

Remarks

The SheetHeight, <u>SheetWidth</u> and <u>DatabaseScale</u> determine the effective area on the drawing in relation to the plotted drawing.

SheetWidth Property

Specifies the width of the sheet.

See Also | Example

Signature

object. Sheet Width

object

DatabasePreferences The object or objects this

property applies to.

SheetWidth

Double; read-write The sheet width.

Remarks

The <u>SheetHeight</u>, SheetWidth and <u>DatabaseScale</u> determine the effective area on the drawing in relation to the plotted drawing.

ShortTangent Property

Returns the short tangent length for compound and simple spirals.

See Also | Example

Signature

object.ShortTangent

object

AlignSpiral The object or objects this property

applies to.

ShortTangent

Double; read-only

The short tangent length.

ShowStartupDialog Property

Toggles the display of the startup dialog.

See Also | Example

Signature

object.ShowStartupDialog

object

<u>PreferencesUser</u> The object or objects this property

applies to.

ShowStartupDialog

Boolean; read-write

TRUE: Displays the startup dialog. FALSE: Hides the startup dialog.

Remarks

The initial value for this property is TRUE. If this property is set to FALSE, the <u>OverrideNew</u> and <u>OverrideOpen</u> properties are also set to FALSE.

You must exit AutoCAD Land Desktop and restart for this change to take effect.

ShowSubfolders Property

Determines whether drawings in subfolders are included.

See Also | Example

Signature

object.ShowSubfolders

object

<u>Drawings</u> The object or objects this property applies

to.

ShowSubfolders

Boolean; read-write

TRUE: Drawings in subfolders are included in the

collection.

FALSE: Drawings in subfolders are not included in

the collection.

Remarks

The <u>Preferences</u> object holds the project level options.

SpeedTablesPath Property

Specifies the directory in which the speed tables are located.

See Also | Example

Signature

object.SpeedTablesPath

object

PreferencesFiles The object or objects this property

applies to.

SpeedTablesPath

String; read-write

The drive and path for the speed table files.

Remarks

By default, this path is C:\Documents and Settings\All Users\Application Data\AutoCAD Land Desktop <version number>\<release number>\Data\Speed Tables. You must exit AutoCAD Land Desktop and restart for this change to take effect.

SpiEasting Property

The Easting coordinate for the spi point.

See Also | Example

Signature

object.SpiEasting

object

AlignSpiral The object or objects this property

applies to.

SpiEasting

Double; read-only

The Easting coordinate for the Spiral Point of

Intersection.

SpilTangent Property

Returns the long tangent length.

See Also | Example

Signature

object.SpilTangent

object

AlignSpiral The object or objects this property

applies to.

SpilTangent

Double; read-only

The long tangent length.

SpiNorthing Property

The Northing coordinate for the spi point.

See Also | Example

Signature

object.SpiNorthing

object

AlignSpiral The object or objects this property

applies to.

SpiNorthing

Double; read-only

The Northing coordinate for the Spiral Point of

Intersection.

SpiralType1 Property

Returns the calculation method of the spiral.

See Also | Example

Signature

object. Spiral Type 1

object

AlignSpiral The object or objects this property

applies to.

SpiralType1

eAeccSpiralType1 enum; read-only

kClothoid: Clothoid spiral

kSinusoid: Sinusoid spiral

kCosinusoid: Cosinusoid spiral

kQuadratic: Quadratic spiral

Remarks

Compound spirals are only supported when using a Clothoid spiral type. No matter what spiral type is set, the command always uses the Clothoid spiral type.

SpiralType2 Property

Returns whether the spiral is a simple, offset, or compound spiral.

See Also | Example

Signature

object.SpiralType2

object

AlignSpiral The object or objects this property

applies to.

SpiralType2

eAeccSpiralType2 enum; read-only

kSimple: Simple spiral

kOffset: Offset spiral

kCompound: Compound spiral

kOffsetCompound: Offset / compound spiral

Remarks

Compound spirals are only supported when using a Clothoid spiral type. No matter what spiral type is set, the command always uses the Clothoid spiral type.

StartDirection Property

The starting direction of the curve or spiral.

See Also | Example

Signature: Overview

1 AlignCurve

1 AlignSpiral

The starting direction of the curve.

Signature

object.StartDirection

object

AlignCurve The object or objects this property

applies to.

StartDirection

Double; read-only

The start direction of the curve.

Remarks

The direction is in radians and measured counter-clockwise from the X axis (East).

The starting direction of the spiral.

Signature

object.StartDirection

object

AlignSpiral

The object or objects this property applies to.

StartDirection

Double; read-only

The direction taken from the TS to the Spi. The angle is in radians and measured counter-clockwise

from the X axis (East).

Remarks

The direction is measured in radians.

StartEasting Property

The Easting coordinate for the beginning of the individual alignment or profile entities.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 ParcelEntity, ParcelCurve, ParcelLine

The Easting coordinate for the beginning of the individual alignment entities.

Signature

object.StartEasting

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

StartEasting

Double; read-only

The Easting coordinate for the beginning point.

The Easting coordinate for the beginning of the individual parcel entities.

Signature

object.StartEasting

object

ParcelEntity, ParcelCurve, ParcelLine

The object or objects this property applies to.

StartEasting

Double; read-only

The Easting coordinate for the beginning point.

StartingStation Property

The object's starting station.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 Alignment

1 ProfileBlock (Civil Engineering Feature)

The StartingStation for individual entities in an alignment.

Signature

object.StartingStation

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

StartingStation

Double; read-only

The starting station for the object.

The StartingStation for the overall alignment.

Signature

object.StartingStation

object

Alignment

The object or objects this property applies to.

StartingStation

Double; read-write

The starting station for the object.

Returns the StartingStation for the Profile.

Signature

object.StartingStation

object

ProfileBlock

The object or objects this property applies to.

StartingStation

Double; read-only The starting station for the Profile.

StartNorthing Property

The Northing coordinate for the beginning of the individual alignment or parcel entities.

See Also | Example

Signature: Overview

1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent

1 ParcelEntity, ParcelCurve, ParcelLine

The Northing coordinate for the beginning of the individual alignment entities.

Signature

object.StartNorthing

object

AlignEntity, AlignCurve, AlignSpiral, AlignTangent

The object or objects this property applies to.

BeginNorthing

Double; read-only

The Northing coordinate for the beginning point.

The Northing coordinate for the beginning of the individual parcel entities.

Signature

object.StartNorthing

object

ParcelEntity, ParcelCurve, ParcelLine

The object or objects this property applies to.

BeginNorthing

Double; read-only

The Northing coordinate for the beginning point.

Station Property (Civil Engineering Feature)

Returns the station for the cross section, cross section block, superelevation, or PVI.

See Also | Example

Signature: Overview

1 <u>CrossSection</u> (Civil Engineering Feature)

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 PVI (Civil Engineering Feature)

1 <u>Superelevation</u> (Civil Engineering Feature)

Returns the station of the cross section.

Signature

object.Station

object

CrossSection The object or objects this property

applies to.

Station

Double; read-only

The station for the cross section.

Returns the station of the cross section block.

Signature

object.Station

object

CrossSectionBlock

The object or objects this property applies to.

Station

Double; read-only

The station for the cross section block.

Returns the Station of the PVI.

Signature

object.Station

object

PVI

The object or objects this property applies to.

Station

Double; read-only

The station for the PVI.

Returns the superelevation station.

Signature

object.Station

object

Superelevation

The object or objects this property applies to.

Station

Double; read-only
The station for the superelevation.

StationAhead Property

The ahead station for the station equation.

See Also | Example

Signature

object. Station Ahead

object

StationEquation The object or objects this property

applies to.

StationAhead

Double; read-write

The ahead station for the station equation.

StationBack Property

The back station for the station equation.

See Also | Example

Signature

object.StationBack

object

StationEquation The object or objects this property

applies to.

StationBack

Double; read-write

The back station for the station equation.

StationEquations Property

The alignment's station equations collection.

See Also | Example

Signature

object.StationEquations

object

Alignment The object or objects this property

applies to.

StationEquations

StationEquations collection; read-only

The alignment station equations collection.

StationElevations Property (Civil Engineering Feature)

An array of stations and elevations for the EGProfile.

See Also | Example

Signature

object.StationElevations

object

EGProfile The object or objects this property applies

to.

StationElevations

Variant (array of doubles); read-write

An array of doubles with the format of Station,

Elevation, Station, etc.

StationIncrement Property (Civil Engineering Feature)

Returns the horizontal distance between vertical grid lines.

See Also | Example

Signature

object.StationIncrement

object

ProfileBlock The object or objects this property

applies to.

StationIncrement

Double; read-only

The Station Increment for the Profile.

Status Property

The status for a surface.

See Also | Example

Signature

object.Status

object

Surface The object or objects this property applies

to.

Status

eAeccSurfaceStatus enum; read-only

kNoData: Surface has no data

kNotBuilt: Surface is not built

kUpToDate: Surface is up to date

kOutOfDate: Surface is out of date

Remarks

The status of a surface will only be accurate when the time between a <u>Build</u> and an addition of Surface Input data is greater than the resolution of the file system.

SuperelevationCode Property (Civil Engineering Feature)

Returns the superelevation code.

See Also | Example

Signature

object. Superel evation Code

object

<u>Superelevation</u> The object or objects this property

applies to.

SuperelevationCode

eAeccSectionSECode enum; read-only

kSEFullCrown: Full crown superelevation

kSEHalfCrown: Half crown superelevation

kSECrownRemoved: Crown removed

kSEFullSuperelevation: Full superelevation

kSEReverseCurve: Reverese curve

kSECompoundCurve: Compound curve

Superelevations Property (Civil Engineering Feature)

Gets the alignment superelevations collection.

See Also | Example

Signature

object.Superelevations

object

Alignment The object or objects this property

applies to.

Superelevations

SuperElevations collection; read-only

The <u>SuperElevations</u> collection.

Surface Property

The collection of surface preferences for the project.

See Also | Example

Signature

object.Surface

object

PreferencesProject The object or objects this

property applies to.

Surface

<u>PreferencesSurface</u>; read-only The <u>PreferencesSurface</u> object.

SurfaceName Property (Civil Engineering Feature)

The name of the EGProfile surface.

See Also | Example

Signature

object.SurfaceName

object

EGProfile The object or objects this property applies

to.

SurfaceName

String; read-only

The name of the EGProfile surface.

Surfaces Property

Gets the Surfaces collection.

See Also | Example

Signature

object.Surfaces

object

Project The object or objects this property applies

to.

Surfaces

Surfaces collection; read-only

The **Surfaces** collection.

Remarks

The <u>Surfaces</u> collection represents all of the <u>Surface</u> objects in this project. This property is valid only for the <u>ActiveProject</u>.

SymbolBlock Property

Specifies the symbol block.

See Also | Example

Signature

object.SymbolBlock

object

DescriptionKey The object or objects this property

applies to.

Symbol Block

String; read-write

Returns and sets the symbol block.

SymbolLayer Property

Specifies the symbol block layer.

See Also | Example

Signature

object.SymbolLayer

object

DescriptionKey The object or objects this property

applies to.

SymbolLayer

String; read-write

Returns and sets the symbol layer.

SymbolManagerPath Property

Specifies the directory in which the symbol sets are located.

See Also | Example

Signature

object.SymbolManagerPath

object

<u>PreferencesFiles</u> The object or objects this property applies to.

SymbolManagerPath

String; read-write
The drive and path for the symbol sets.

Remarks

By default, this path is C:\Documents and Settings\All Users\Application Data\AutoCAD Land Desktop <version number>\<release number>\Data\Symbol Manager. You must exit AutoCAD Land Desktop and restart for this change to take effect.

SystemPath Property

Specifies the directory in which AutoCAD Land Desktop is installed.

See Also | Example

Signature

object.SystemPath

object

<u>PreferencesFiles</u> The object or objects this property

applies to.

SystemPath

String; read-only

The drive and path where AutoCAD Land Desktop

is installed.

TangentLabelIncrement Property (Civil Engineering Feature)

Returns the distance between tangent elevation labels.

See Also | Example

Signature

object.TangentLabelIncrement

object

<u>ProfileBlock</u> The object or objects this property applies to.

Tangent Label Increment

Double; read-only

The Tangent Labeling Increment for the Profile.

TangentLength Property

The tangent length of the curve.

See Also | Example

Signature: Overview

1 AlignCurve

1 ParcelCurve

The tangent length of the alignment curve.

Signature

object.TangentLength

object

AlignCurve The object or objects this property

applies to.

TangentLength

Double; read-only

The tangent length of the curve.

The tangent length of the parcel curve.

Signature

object.TangentLength

object

ParcelCurve

The object or objects this property applies to.

TangentLength

Double; read-only
The tangent length of the curve.

TempPath Property

Specifies the directory in which the temporary files are stored.

See Also | Example

Signature

object.TempPath

object

PreferencesFiles The object or objects this property

applies to.

TempPath

String; read-write

The drive and path for the temporary files.

Remarks

By default, this path is c:\temp. You must exit AutoCAD Land Desktop and restart for this change to take effect.

TextAbove Property

The text above the object.

See Also | Example

Signature

object.TextAbove

object

AeccCurveText The object or objects this property

applies to.

TextAbove

String; read-write

The text above the curve text object.

TextBelow Property

The text below the object.

See Also | Example

Signature

object.TextBelow

object

AeccCurveText The object or objects this property

applies to.

TextBelow

String; read-write

The text below the curve text object.

TextHeight Property

The height of the text in the drawing.

See Also | Example

Signature

object.TextHeight

object

DatabasePreferences The object or objects this

property applies to.

TextHeight

Double; read-write

The text height in inches or millimeters.

Remarks

The default value is 0.125 inches, or 3.175 millimeters, depending on the setting of the MeasurementUnit property.

TextOffsetAbove Property

The offset of the text above the object.

See Also | Example

Signature

object. Text Off set Above

object

AeccCurveText The object or objects this property

applies to.

TextOffsetAbove

Double; read-write

The offset of the text above the curve text object.

TextOffsetBelow Property

The offset of the text below the object.

See Also | Example

Signature

object.TextOffsetBelow

object

AeccCurveText The object or objects this property

applies to.

TextOffsetBelow

Double; read-write

The offset of the text below the curve text object.

TextSize Property

The size of the text above and below the object.

See Also | Example

Signature

object.TextSize

object

AeccCurveText The object or objects this property

applies to.

TextSize

Double; read-write

The size of the text above and below the curve text

object.

ThetaExt Property

Returns the external theta for a compound spiral.

See Also | Example

Signature

object.ThetaExt

object

AlignSpiral The object or objects this property

applies to.

ThetaExt

Double; read-only

The external theta for a compound spiral.

Time Property

The lock creation date and time.

See Also | Example

Signature

object.Time

object

FileLock The object or objects this property applies

to.

Time

String; read-only

The date and time of the file lock.

TinPoints Property

Returns the TinPoints collection.

See Also | Example

Signature

object.TinPoints

object

SurfaceOutputs The object or objects this property

applies to.

TinPoints

TinPoints collection; read-only

The <u>TinPoints</u> collection.

TotalX Property

The total X value for the spiral.

See Also | Example

Signature

object.TotalX

object

AlignSpiral The object or objects this property

applies to.

TotalX

Double; read-only

Total X value.

TotalY Property

The total Y value for the spiral.

See Also | Example

Signature

object.TotalY

object

AlignSpiral The object or objects this property

applies to.

TotalY

Double; read-only

Total Y value.

Type Property

The type properties for Objects.

See Also | Example

Signature: Overview

- 1 AlignCurve, AlignEntity, AlignSpiral, AlignTangent
- l **Boundary**
- 1 BreakLine
- 1 <u>CrossSectionSurface</u> (Civil Engineering Feature)
- l <u>EGProfile</u> (Civil Engineering Feature)
- 1 FGProfile (Civil Engineering Feature)
- 1 ParcelCurve, ParcelLine
- 1 StationEquation
- 1 Surface
- 1 Watershed

The type of the alignment entity.

Signature

object.type

object

AlignCurve, AlignEntity, AlignSpiral, AlignTangent

The object this property applies to.

Type

eAeccAlignEntityType enum; read-only

The type of the Alignment.

kCurve: Alignment Curve

kSpiral: Alignment Spiral

kTangent: Alignment Tangent

The type of the Boundary.

Signature

object.Type

object

Boundary

The object or objects this property applies to.

Type

eAeccBoundaryType enum; read-only

The type of the Boundary.

kBoundaryTypeShow: Boundary is visible

kBoundaryTypeHide: Boundary is hidden

kBoundaryTypeOuter:

Boundary is the outer

most boundary

kBoundaryTypeUnInit:

Boundaries are not initalized

Remarks

A Boundary can not be changed to an outer (kBoundaryTypeOuter) boundary. If the Boundary is an outer, it is always visible.

The type of the BreakLine.

Signature

object.Type

object

BreakLine

The object or objects this property applies to.

Type

eAeccBreakLineType enum; read-only The type of the BreakLine.

kStandard: Standard breakline

kProximity: Proximity breakline

kWallLeft: Left wall breakline

kWallRight: Right wall breakline

kNonDestructive: Nondestructive breakline

Returns the type of the cross section surface.

Signature

Object.Type

object

<u>CrossSectionSurface</u>

The object or objects this property applies to.

Type

 $eAeccCrossSectionSurfaceType\ enum;\ read-only$

The type od cross section surface.

kExistingGround: 0

kTopSurface: 10

kDatumSurface: 20

kTemplateSurface: 30

kSubassemblySurface: 40

kMatchSurface: 50

Returns the type of existing ground profile.

Signature

Object.Type

object

EGProfile

The object or objects this property applies to.

Type

eAeccEGProfileType enum; read-only

The VerticalOffset type.

kEgCenter: Existing ground center type

kEgLeft: Existing ground left type

kEgNone: Existing ground none type

kEgRight: Existing ground right type

Returns the type of the Finished Ground Profile.

Signature

Object.Type

object

FGProfile

The object or objects this property applies to.

Type

eAeccFGProfileType enum; read-only

Returns the type of the Finished Ground Profile.

kFgCenter: Finished ground profile center

type

kFgDitchLeftt: Finished ground ditch left type

kFgDitchRight: Finished ground ditch right type

kFgLeft1: Finished ground left 1 trans type

kFgLeft2: Finished ground left 2 trans type

kFgLeft3: Finished ground left 3 trans type

kFgLeft4: Finished ground left 4 trans type

kFgLeft5: Finished ground left 5 trans type

kFgLeft6: Finished ground left 6 trans type

kFgLeft7: Finished ground left 7 trans type

kFgLeft8: Finished ground left 8 trans type

kFgNone: Finished ground none type

kFgRight1: Finished groung right 1 trans type

kFgRight2: Finished groung right 2 trans type

kFgRight3: Finished groung right 3 trans type

kFgRight4: Finished groung right 4 trans type

kFgRight5: Finished groung right 5 trans type

kFgRight6: Finished groung right 6 trans type

kFgRight7: Finished groung right 7 trans type

kFgRight8: Finished groung right 8 trans type

Returns the parcel entity type.

Signature

object.Type

object

ParcelCurve, ParcelLine

The object or objects this property applies to.

Type

eAeccParcelEntityType enum; read-only

The type of the parcel entity.

kParcelCurve: Parcel entity is a curve

kParcelLine: Parcel entity is a line

The type of station equation.

Signature

object.Type

object

StationEquation

The object or objects this property applies to.

Type

eAeccStationEquationType enum; read-write

The type of the Station Equation.

kIncreasing: Station Equation increases

stationing

kDecreasing: Station Equation decreases

stationing

The Surface type.

Signature

object.Type

object

Surface

The object or objects this property applies to.

Type

eAeccSurfaceType enum; read-only

The type of the Surface.

kUnknownSurface: Unknown surface type

kTin: Tin surface type

kCompositeVolume: Composite volume surface

type

kGridVolume: Grid volume surface type

The type of the WaterShed.

Signature

object.Type

object

WaterShed

The object or objects this property applies to.

Type

eAeccWaterShedType enum: read-only

The type of the WaterShed.

A watershed that includes a

kBoundaryPoint: boundary point, the point where the channel of water

drains off the surface

A watershed that includes

kBoundarySegment: part of the boundary of a

surface as its drain target

kLocalMin: The watershed is a

depression

A flat area where the water

collects, passes over, or

causes a channel to split. An

ambiguous watershed.

A watershed where the water

can go to more than one

drain target. An ambiguous

watershed

kNonBoundary:

kMultiDrain:

kFlatArea:

A watershed formed by a

kUnknownSink: notch and which drains to

two or more drain targets

Remarks

Each watershed subarea that you delineate is categorized based on drain target.

UpperRight Property (Civil Engineering Feature)

Returns the Upper Right Coordinates of the ProfileBlock as XY.

See Also | Example

Signature

object.UpperRight

object

<u>ProfileBlock</u> The object or objects this property

applies to.

UpperRight

Variant (two-element array of doubles); read-only An array of the X and Y value, representing the upper right coordinates of a profile. This and the coordinates of the origin define the drawing limits of a profile.

UsedPointNumbers Property

Returns the list of used point numbers in the current project database.

See Also | Example

Signature

object.UsedPointNumbers()

object

<u>CogoPoints</u> The object or objects this property

applies to.

UsedPointNumbers

String; read-only

Comma delimited string with all used points with

groups separated by hyphens.

Remarks

Typical examples of the point string format are: "1,2,3,5,10" "1-100,1000-2000"

User Property

Gets the PreferencesUser object.

See Also | Example

Signature

object.User

object

Preferences The object or objects this property

applies to.

User

<u>PreferencesUser</u> object; read-only Gets the <u>PreferencesUser</u> object.

Remarks

This object specifies the options maintained on a per-user basis.

Utility Property

Gets the Utility object.

See Also | Example

Signature

object.Utility

object

Document The object or objects this property

applies to.

Utility

Utility object; read-only

The Utility object.

VerticalScale Property

Specifies the scale at which the vertical axis of the drawing is displayed, or the vertical scale for alignment profiles and cross sections.

See Also | Example

Signature: Overview

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 DatabasePreferences

1 ProfileBlock (Civil Engineering Feature)

Specifies the vertical scale at which the alignment cross section is drawn.

Signature

object.VerticalScale

object

CrossSectionBlock The object or objects this

property applies to.

VerticalScale

Double; read-only

The vertical scale at which the alignment cross

section is drawn.

Specifies the scale at which the vertical axis of the drawing is displayed.

Signature

object.VerticalScale

object

DatabasePreferences

The object or objects this property applies to.

VerticalScale

Double; read-write

The vertical scale. For example, for a 1:200 scale,

type 200.

Remarks

The VerticalScale is compared against the <u>DatabaseScale</u> to calculate the vertical exaggeration in profiles and cross sections. It does not actually change the scale that is used when the drawing is plotted.

Specifies the vertical scale at which the alignment profiles are drawn.

Signature

object.VerticalScale

object

ProfileBlock

The object or objects this property applies to.

VerticalScale

Double; read-only

The vertical scale at which the profile is drawn.

Volume Property

The volume of the surface.

See Also | Example

Signature

object.Volume

object

Surface The object or objects this property applies

to.

Volume

Double; read-only The surface volume.

Remarks

This is an Extended Statistic.

VolumeDisplayUnit Property

The units used to display volumes.

See Also | Example

Signature

object.VolumeDisplayUnit

object

<u>DatabasePreferences</u> The object or objects this

property applies to.

VolumeDisplayUnit

AecBuiltInUnit enum; read-write

aecUnitCubicInch: Cubic inch unit

aecUnitCubicFoot: Cubic foot unit

aecUnitCubicYard: Cubic yard unit

aecUnitCubicMillimeter: Cubic millimeter unit

aecUnitCubicCentimeter: Cubic centimeter unit

aecUnitCubicMeter: Cubic meter unit

VolumePrecision Property

The precision used to display volumes.

See Also | Example

Signature

object.VolumePrecision

object

DatabasePreferences The object or objects this

property applies to.

VolumePrecision

Long; read-write

The display precision used for volumes.

Remarks

The precision settings are used only for labeling and listing values, not actual computations. AutoCAD Land Desktop commands always calculate all numbers up to the highest internal precision.

VolumeSuffix Property

The suffix displayed after an volume.

See Also | Example

Signature

object.VolumeSuffix

object

DatabasePreferences The object or objects this

property applies to.

VolumeSuffix

String; read-write

The suffix displayed after a volume.

Remarks

The default value is "Cu.Ft." or "m3", depending on the value of the MeasurementUnit property. The value must 255 characters or less in length.

Watersheds Property

Returns the WaterSheds collection

See Also | Example

Signature

object.WaterSheds

object

SurfaceOutputs The object or objects this property

applies to.

WaterSheds

WaterSheds collection; read-only

The Watersheds collection.

Objects



Α

AeccApplication An instance of the AutoCAD Land Desktop application.

<u>AeccContour</u> The AeccContour represents a contour entity in the document.

<u>AeccContourStyle</u> The AeccContourStyle represents a contour style entry in the dictionary.

<u>AeccCurveText</u> The AeccCurveText represents a curve text entity in the document.

AeccPoint The AeccPoint represents a cogo point entity in the document.

<u>AlignCurve</u> The AlignCurve object represents a curve entity for the alignment.

<u>AlignEntities</u> The AlignEntities collection represents all of the entities for an Alignment.

AlignEntity The AlignEntity object is the base class for AlignTangent, AlignCurve and AlignSpiral.

<u>Alignment</u> The Alignment object represents an alignment in the project database.

<u>Alignments</u> The Alignments collection represents all of the Alignment objects in the project database.

<u>AlignSpiral</u> The AlignSpiral object represents a spiral entity for the alignment.

<u>AlignTangent</u> The AlignTangent object represents a tangent entity for the alignment.

<u>Boundaries</u> The Boundaries collection represents all of the boundaries for the Surface.

Boundary The Boundary object represents a single boundary for the Surface.

BreakLine The BreakLine object represents a single breakline for the Surface.

<u>BreakLines</u> The BreakLines collection represents all of the breaklines for the Surface.

C

CogoPoint The CogoPoint represents a COGO point in the project database.

<u>CogoPoints</u> The CogoPoints collection represents all of the COGO points in the project database.

<u>ContourItem</u> The ContourItem object represents a single contour for the ContourItems collection.

<u>ContourItems</u> The ContourItems collection represents all of the input contours for the Surface.

<u>CrossSection</u> (Civil Engineering Feature) The CrossSection object represents a single section for an alignment

<u>CrossSectionBlock</u> (Civil Engineering Feature) The CrossSectionBlock object represents an alignment cross section in the current drawing.

<u>CrossSectionBlocks</u> (Civil Engineering Feature) The CrossSectionBlocks collection represents all of the alignment cross sections that are in the current drawing.

<u>CrossSectionPointCode</u> (Civil Engineering Feature) The CrossSectionPointCode object represents a single point code for a cross

section.

<u>CrossSectionPointCodes</u> (Civil Engineering Feature) The CrossSectionPointCodes collection represents all of the point codes for a cross section.

<u>CrossSections</u> (Civil Engineering Feature) The CrossSections collection represents all of the cross sections for an alignment.

<u>CrossSectionSurface</u> (Civil Engineering Feature) The CrossSectionSurface object represents a single surface section for a cross section.

<u>CrossSectionSurfaces</u> (Civil Engineering Feature) The CrossSectionSurfaces collection represents all of the surfaces for a cross section.

D

<u>DatabasePreferences</u> This object specifies the current AutoCAD Land Desktop drawing specific settings.

<u>DEMFile</u> The DEMFile object represents a single Digital Elevation Model file for the Surface.

<u>DEMFiles</u> The DEMFiles collection represents all of the Ditial Elevation Model files for the Surface.

<u>DescriptionKey</u> The DescriptionKey object represents the description key for point.

<u>DescriptionKeyFile</u> The DescriptionKeyFile collection represents all of the description keys in a project description key file. Each key is contained in a particular description key file.

<u>DescriptionKeyFiles</u> The DescriptionKeyFiles collection represents all of the description keys for a project description key file.

Document An AutoCAD Land Desktop drawing.

<u>Documents</u> The collection of all AutoCAD Land Desktop drawings open in the current session.

Drawing An AutoCAD Land Desktop project-based drawing.

<u>Drawings</u> The collection of all AutoCAD Land Desktop drawings in a project.

Ε

Edge The Edge object represents a single output edge for the Surface.

<u>Edges</u> The Edges collection represents all of the output edges for the Surface.

<u>EGProfile</u> (Civil Engineering Feature) The EGProfile object represents the station / elevation information for a given surface.

<u>EGProfiles</u> (Civil Engineering Feature) The EGProfiles collection represents all of the existing ground vertical profiles.

<u>ElevationContour</u> The ElevationContour object represents a single contour for the ElevationContours collection.

<u>ElevationContours</u> The ElevationContours collection represents all of the calculated contours at a given elevation for the Surface.

F

<u>Face</u> The face object represents a single calculated face (triangle) for the Surface.

<u>Faces</u> The faces collection represents all of the calculated faces (triangles) for the Surface.

<u>FGProfile</u> (Civil Engineering Feature) The FGProfile object represents a finished ground vertical PVIs.

<u>FGProfiles</u> (Civil Engineering Feature) The FGProfiles collection represents all of the finished grade profiles in the project.

FileLocks The collection of all file locks for a particular project. G Н J K L M Ν 0 Р Parcel The Parcel object represents a parcel in the current project database. ParcelCurve The ParcelCurve object represents a curve entity for the parcel. ParcelEntities TheParcelEntities collection represents all of the entities for a Parcel.

FileLock A file lock in the AutoCAD Land Desktop project.

<u>ParcelLine</u> The ParcelLine object represents a line entity for the Parcel.

and ParcelLine.

ParcelEntity The ParcelEntity collection is the base class for ParcelCurve

<u>Parcels</u> The Parcels collection represents all of the parcels that are in the current project.

<u>PointFile</u> The PointFile object represents a single point file for the Surface.

<u>PointFiles</u> The PointFiles collection represents all of the point files for the Surface.

<u>PointGroup</u> The PointGroup collection represents a Point Group in the project database.

<u>PointGroupName</u> The PointGroupName object represents a single PointGroupName for the PointGroupNames collection.

<u>PointGroupNames</u> The PointGroupNames collection represents all of the PointGroupNames for the Surface.

<u>PointGroups</u> The PointGroups collection represents all of the Point Groups in the project database.

<u>Preferences</u> This object specifies the current AutoCAD Land Desktop settings.

<u>PreferencesAlignment</u> This object specifies the settings for Horizontal Alignments.

<u>PreferencesCogo</u> This object specifies the settings for Points.

<u>PreferencesCrossSection</u> (Civil Engineering Feature) This object specifies the settings for Alignment Cross Sections.

<u>PreferencesFiles</u> This object specifies the paths used by the AutoCAD Land Desktop.

<u>PreferencesParcel</u> This object specifies the settings for Parcels.

<u>PreferencesProfile</u> (Civil Engineering Feature) This object specifies the settings for Alignment Vertical Profiles.

<u>PreferencesProject</u> This object specifies the project settings.

<u>PreferencesSurface</u> This object specifies the settings for Surfaces.

<u>PreferencesUser</u> This object specifies the options maintained on a peruser basis.

<u>ProfileBlock</u> (Civil Engineering Feature) The ProfileBlock object represents a profile in the current drawing.

<u>ProfileBlocks</u> (Civil Engineering Feature) The ProfileBlocks collection represents all of the profiles that are in the current drawing.

Project An AutoCAD Land Desktop project.

<u>Projects</u> The collection of all AutoCAD Land Desktop projects on the network.

Prototype An AutoCAD Land Desktop project prototype.

<u>Prototypes</u> The collection of all AutoCAD Land Desktop project prototypes.

PVI (Civil Engineering Feature) The PVI object represents a Point of Vertical Interstection for a Finished Ground Profile.

<u>PVIs</u> (Civil Engineering Feature) The PVIs collection represents all of the Point of Vertical Intesection objects in the project.

Q

R

S

<u>StationEquation</u> The StationEquation object represents a station equation for the alignment.

<u>StationEquations</u> The StationEquations collection represents all of the station equations for the Alignment.

<u>Superelevation</u> (Civil Engineering Feature) The Superelevation object represents a single superelevation for an alignment.

Superelevations (Civil Engineering Feature) The Superelevations

collection represents all of the superelevations for a cross section.

<u>Surface</u> The Surface object represents a Surface in the project database.

<u>SurfaceInputs</u> The SurfaceInputs object represents all inputs for a Surface.

<u>SurfaceOutputs</u> The SurfaceOutputs object represents all outputs for a Surface.

<u>Surfaces</u> The Surfaces collection represents all of the Surface objects in the project database.

Т

<u>TinPoint</u> The TinPoint represents a calculated point for the Surface.

<u>TinPoints</u> The TinPoints represents all of the calculated points for the Surface.

U

<u>Utility</u> A series of methods provided for utility purposes.

V

W

<u>WaterShed</u> The WaterSheds collection represents all of the WaterShed items for the Surface.

<u>WaterSheds</u> The WaterShed object represents a single watershed item for the WaterSheds collection.

X

Υ

Ζ

1 ■ AeccApplication Object

An instance of the AutoCAD Land Desktop application.

VBA object name: AeccApplication

For VB:

GetObject("Acad.Application") or

reateObject("Acad.Application") then

Create using: etInterfaceObject("Aecc.Application")

For AutoCAD VBA:

not applicable. The application is always available.

Access via: <u>Application</u> Property

The properties associated with the AeccApplication object reflect the properties of the main application window. The methods control the loading or listing of the currently loaded external applications and interface objects.

The active document (AutoCAD Land Desktop drawing) can be accessed using the <u>ActiveDocument</u> property. The active project (AutoCAD Land

Desktop project) can be accessed using the ActiveProject property.

See the <u>Application</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Properties

ActiveDocument

ActiveProject

Methods Application

Events

None **Documents**

None

Preferences

Projects

Prototypes

↑ AeccContour Object

The AeccContour represents a contour entity in the document.

VBA object name: AeccContour

Create using: AcadModelSpace.AddCustomObject("AeccDbContour")

Access via: AcadModelSpace.Item

In addition to the properties listed below, the AeccContour object supports all of the members of the <u>AcadEntity</u> object.

Properties

Application

Methods

ContourStyle

AddLabelAt

<u>ContourStyleName</u> Events

RemoveAllLabels

None

Coordinates

RemovelabelAt

Florestion

<u> Lievaliuii</u>

LabelPoints

↑ ↓ AeccContourStyle

Object

The AeccContourStyle represents a contour style entry in the dictionary.

VBA object AeccContourStyle

Create using: N/A

Access via: ThisDrawing.Dictionaries("AECC_CONTOUR_STYLES").Item

In addition to the properties listed below, the AeccContourStyle object supports all of the members of the <u>AcadObject</u> object.

The contour styles dictionary is not added to the drawing until a contour object is created, or when the contour style manager is used. The "Standard" contour style is created by default.



↑ AeccCurveText Object

The AeccCurveText represents a curve text entity in the document.

VBA object name: AeccCurveText

Create using: AcadModelSpace.AddCustomObject("AecDbCurveText")

Access via: AcadModelSpace.Item

In addition to the properties listed below, the AeccCurveText object supports all of the members of the <u>AcadEntity</u> object.

Properties

Application

<u>TextAbove</u>

Methods <u>TextBelow</u> Events

<u>SetReferenceCurve</u> None

TextOffsetAbove

Toy+Offco+Dolow

<u>IEXIOIISEIDEIUW</u>

<u>TextSize</u>

↑ AeccPoint Object

The AeccPoint represents a cogo point entity in the document.

VBA object name: AeccPoint

Create using: AcadModelSpace.AddCustomObject("AeccDbPoint;")

Access via: AcadModelSpace.Item

In order to have AeccPoints visible in the drawing using AddCustomObject() and setting properties, the Number property is required to be set.

In addition to the properties listed below, the AeccPoint object supports all of the members of the AcadEntity object.

Application

Description

Methods Events

<u> ⊨asung</u>

None None

Elevation

Northing

Number

↑ AlignCurve Object

The AlignCurve object represents a curve entity for the alignment.

VBA object name: AeccAlignCurve

Create using: Alignment.AddCurve

Access via: AlignEntities.Item

Properties

Application

CCWFlag

CenterEasting

CenterNorthing

ChordDirection

ChordLength

<u>Delta</u> **EndDirection EndEasting EndingStation** Methods EndNorthing **Events** None ExternalSecant Modified **Length MidOrdinate** <u>ObjectID</u> **PiEasting PiNorthing Radius StartDirection StartEasting StartingStation StartNorthing**

TangentLength

Type

↑ AlignEntities Collection

The AlignEntities collection represents all of the entities for an Alignment.

VBA object name: AeccAlignEntities

Create using: N/A

Access via: Alignment. AlignEntities

This collection returns AlignTangent, AlignCurve and AlignSpiral objects. When you access a member of the collection using the Item method or the For Each statement, assign the result to a VARIANT. You can then use the TypeOf function to determine which object type (AlignTangent, AlignCurve or AlignSpiral) to assign it to.

Methods
Application

Item
Count

Properties

Events

Modified

↑ AlignEntity Collection

The AlignEntity object is the base class for AlignTangent, AlignCurve and AlignSpiral.

VBA object name: AeccAlignEntity

Create using: N/A

Access via: AlignEntities.Item

All alignment entities are derived from AlignEntity - they all share the methods, properties and events listed below. You can use this base class as the object type returned from the method Item of the AlignEntities collection. If you need access to the complete details of an alignment entity, use the Type property (or TypeOf function in VB) to determine whether to assign the returned object to an AlignCurve or AlignCurve or AlignSpiral.

Properties

Application

EndEasting

EndingStation

EndNorthing

Methods Length

Events

None

Modified

<u>ObjectID</u>

StartEasting

StartingStation

StartNorthing

<u>Type</u>

↑ Alignment Object

The Alignment object represents an Alignment in the project database.

VBA object name: AeccAlignment

Create using: Alignments.Add

Access via: Alignments.Item

The Alignment object exposes alignment geometry through the <u>AlignEntities</u> collection. Each AlignEntity (an <u>AlignTangent</u>, <u>AlignCurve</u> or <u>AlignSpiral</u>) exposes geometric properties, as well as the <u>ObjectID</u> of the corresponding entity in the current drawing (if any).

Valid alignment ObjectIDs are defined during an Import. The ObjectIDs are valid until the alignment is saved. Alignments are automatically saved each time an alignment property is modified when AutoSave is set to true, or when an alignment Save is executed.

	Properties
Methods	
	<u>AlignEntities</u>

<u>AddCurve</u> <u>Application</u>

<u>AddSpiral</u> <u>CrossSections</u> (Civil Engineering

Feature)

<u>AddTangent</u>

Description

ExternalStaToInternal

EGProfiles(Civil Engineering Feature)

<u>GetStaStrWithEquations</u>

EndingStation

<u>GetStaWithEquations</u>

FGProfiles (Civil Engineering Feature) Modified

Events

<u>Import</u>

LockType

LineIntersection

Name

PerpIntersection

Number

PointLocation

StartingStation

RemoveAll

StationEquations

Save

Superelevations (Civil Engineering

<u>StationOffset</u> Feature)

↑ Alignments Collection

The Alignments collection represents all of the Alignment objects in the project database.

VBA object name: AeccAlignments

Create using: N/A

Access via: Project. Alignments

All Alignments except for the current Alignment are loaded with read only permissions. The current alignment can have read or read-write permissions. If no other user has control of the current alignment, the read-write permissions will be given. If another user has read-write permission, then only read permission will be given. As you switch the current Alignment, permissions will be updated. Also, by setting the current Alignment, you are also setting the current Alignment for the AutoCAD Land Desktop menus.

Methods	
Add	Properties

Application

Count

<u>AlignmentFromObjectID</u>

<u>AutoSave</u>

<u>Delete</u>

Modified

<u>DoubleToStaFormat</u>

CurrentAlignment

<u>ltem</u>

↑ AlignSpiral Object

The AlignSpiral object represents a spiral entity for the alignment.

VBA object name: AeccAlignSpiral

Create using: Alignment.AddSpiral

Access via: AlignEntities.Item

Properties

 $\underline{\mathsf{A}}$

AD1

AD2

Application

BeginCondition

Delta

```
EndEasting
      EndingStation
      EndNorthing
      ExtEasting
      ExtNorthing
      <u>K</u>
      Length
      LExt
Methods LOffset
                       Events
None ObjectID
                      Modified
      <u>P</u>
      RadialDistance
      ShortTangent
      SpiEasting
      SpilTangent
      SpiNorthing
      SpiralType1
      SpiralType2
      StartDirection
      StartEasting
```

StartingStation

StartNorthing

ThetaExt

<u>TotalX</u>

<u>TotalY</u>

<u>Type</u>

↑ AlignTangent Object

The AlignTangent object represents a tangent entity for the alignment.

VBA object name: AeccAlignTangent

Create using: Alignment.AddTangent

Access via: AlignEntities.Item

Properties

Application

Direction

EndEasting

EndingStation

Methods EndNorthing

Events

None <u>Length</u> <u>Modified</u>

<u>ObjectID</u>

StartEasting

<u>StartingStation</u>

StartNorthing

<u>Type</u>

↑ ■ Boundaries Collection

The Boundaries collection represents all of the boundaries for the current Surface.

VBA object name: AeccBoundaries

Create using: N/A

Access via: SurfaceInputs.Boundaries

There are three types of boundaries: outer, hidden, and visible. If an outer boundary exists for the Surface, it must be the first boundary and it is visible. Also, a boundary can not be changed from a hidden / visible boundary to an outer boundary.

Methods Properties

Add Application

Events

Delete Count Modified

<u>Item VerticalScale</u>

1 ■ Boundary Object

The Boundary object represents a single boundary for the Surface.

VBA object name: AeccBoundary

Boundaries.Add **Create using:**

Access via: Boundaries.Item

Properties

Application

Coordinates

Methods Description

Events

None

Modified

<u>Id</u>

<u>IsBreakLine</u>

Type

↑ BreakLine Object

The BreakLine object represents a single breakline for the Surface.

VBA object name: AeccBreakLines

Create using: BreakLines.Add

Access via: BreakLines.Item

Properties

Application

Methods Coordinates Events

None <u>Description</u> <u>Modified</u>

<u>Id</u>

Type

↑ ▶ BreakLines Collection

The BreakLines collection represents all of the breaklines for the Surface.

VBA object name: AeccBreakLines

Create using: N/A

Access via: SurfaceInputs.BreakLines

Methods

Properties

<u>Application</u> Events

<u>Delete</u> <u>Modified</u>

Count

<u>ltem</u>

↑ U CogoPoint Object

The CogoPoint represents a COGO point in the project database.

VBA object name: AeccCogoPoint

Create using: CogoPoints.Add

Access via: CogoPoints.Item

Properties

Application

Coordinates

Easting

Elevation

FullDescription

GridEasting

GridNorthing

GroupName

Methods Latitude

<u>Save</u>

Modified

Events

LockType

Longitude

<u>Name</u>

Northing

<u>Number</u>

OverrideDescription

OverrideElevation

OverrideName

RawDescription

↑ CogoPoints Collection

The CogoPoints collection represents all of the COGO points in the project database.

VBA object name: AeccCogoPoints

Create using: N/A

Access via: Project.CogoPoints

The CogoPoints collection represents all Cogo points for the current project database. There exists only one CogoPoints collection for at a time.

Methods	
Add	Properties
	<u>Application</u>
ArrayToPointString	<u>AutoSave</u>
<u>Delete</u>	Count

<u>item</u>

IsNameSupported

Modified

Events

LockPoints

LockedPointNumbers

PointByNumber

<u>NextPointNumber</u>

<u>PointNumberFromObjID</u>

PointNameSize

<u>PointStringToArray</u> <u>UsedPointNumbers</u>

UnlockPoints

↑ ↓ ContourItem Object

The ContourItem object represents a single contour for the ContourItems collection.

VBA object name: AeccContourItem

Create using: ContourItems.Add

Access via: ContourItems.Item

Properties

Methods Application Events

None Coordinates Modifed

<u>Id</u>

↑ Use ContourItemsCollection

The ContourItems collection represents all of the input contours for the Surface.

VBA object name: AeccContourItems

Create using: N/A

Access via: SurfaceInputs.ContourItems

Methods

Delete

Add **Properties**

Application Events

Modified

Count

<u>Item</u>

(Civil Engineering Feature)

The CrossSection object represents a single section for an alignment.

VBA object name: AeccCrossSection

Create using: N/A

Access via: CrossSections.Item

Properties

Application

CrossSectionPointCodes

Methods CrossSectionSurfaces

Import MaxElevation Events

Modified

SectionVolume MaxOffset

MinLlevation

MinOffset

Station

Object (Civil Engineering Feature)

The CrossSectionBlock object represents an alignment cross section in the current drawing.

VBA object name: AeccCrossSectionBlock

Create using: N/A

Access via: CrossSectionBlocks.Item

Properties

Application

CenterlineOffset

Coordinates

DatumElevation

Methods

EGPrecision

_

<u>GetLayerName</u>

FGPrecision

Events

OffsetElevationToXy

Modified

XyToOffsetElevation

LeftWidth

<u>Name</u>

<u>Height</u>

RightWidth

Station

VerticalScale

The CrossSectionBlocks collection represents all of the alignment cross sections that are in the current drawing.

VBA object name: AeccCrossSectionBlocks

Create using: N/A

Access via: Document.CrossSectionBlocks

 $\frac{\text{Methods}}{\text{Application}} \frac{\text{Events}}{\text{Modified}}$ $\frac{\text{Count}}{\text{Count}}$

The CrossSectionPointCode object represents a single point code for a cross section.

VBA object name: AeccCrossSectionPointCode

Create using: N/A

Access via: CrossSectionPointCodes.Item

Properties

Application

Methods Coue

Events

None <u>Description Modified</u>

Elevation

Offset

CrossSectionPointCodes Collection (Civil Engineering Feature)

The CrossSectionPointCodes collection represents all of the point codes for a cross section.

VBA object name: AeccCrossSectionPointCodes

Create using: N/A

Access via: CrossSection.CrossSectionPointCodes

 $\frac{\text{Methods}}{\text{Application}} \frac{\text{Events}}{\text{Modified}}$ $\frac{\text{Count}}{\text{Count}}$

(Civil Engineering Feature)

The CrossSections collection represents all of the cross sections for an alignment.

Events

VBA object name: AeccCrossSections

Create using: N/A

Access via: Alignments.CrossSections

Methods

<u>Properties</u>

Application

PointCodeDescription Modified

Count

SectionByStation

↑ CrossSectionSurface

Object (Civil Engineering Feature)

The CrossSectionSurface object represents a single surface section for a cross section.

VBA object name: AeccCrossSectionSurface

Create using: N/A

Access via: CrossSectionSurfaces.Item

Properties

Application

Methods Name

Events

None <u>Modified</u>

OffsetElevations

Type

CrossSectionSurfaces Collection (Civil Engineering Feature)

The CrossSectionSurfaces collection represents all of the surfaces for a cross section.

VBA object name: AeccCrossSectionSurfaces

Create using: N/A

Access via: CrossSection.CrossSectionSurfaces

 $\frac{\text{Methods}}{\text{Application}} \frac{\text{Events}}{\text{Modified}}$ $\frac{\text{Count}}{\text{Count}}$

↑ ↓ DatabasePreferences

Object

This object specifies the current AutoCAD Land Desktop drawing specific settings.

VBA object name: AeccDatabasePreferences

Create using: N/A

Access via: Document Preferences

Database preferences represent all the options that reside in a drawing. These include settings in the AdCADD block, dictionary and the AecBaseSetup object.

The DatabasePreferences object can be referenced from the Preferences property on the Document object.

See the <u>DatabasePreferences</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Properties

AngularAzimuth

AngularDisplayFormat

AngularPrecision

Application

AreaDisplayUnit

AreaPrecision

AreaSuffix

BasePoint

BasePointNE

BorderBlockFilename

BorderBottomMargin

BorderLeftMargin

BorderLineWidth

BorderRightMargin

<u>BorderStyle</u>

Methods

BorderTopMargin

ConvertToCurrentAreaDisplay

CoordinatePrecision

ConvertToCurrentVolumeDisplay

<u>CoordinateZone</u>

Events

<u>LoadSetupProfile</u>

DatabaseScale

None

SaveAsDefault

<u>SaveSetupProfile</u>

ElevationPrecision

FacetDeviation

LayerFile

LayerStandard

LinearDisplayFormat

LinearPrecision

LinearUnit

MeasurementUnit

NorthRotation

ProjectName

ScaleOnInsert

SheetHeight

SheetWidth

TextHeight

VerticalScale

VolumeDisplayUnit

VolumePrecision

VolumeSuffix

↑ DEMFile Object

The DEMFile object represents a single Digital Elevation Model file for the Surface.

VBA object name: AeccDEMFile

Create using: DEMFiles.Add

Access via: DEMFiles.Item

Properties

Methods Application

Events

None CoordinateZone Modified

Name

↑ DEMFiles Collection

The DEMFiles collection represents all of the Ditial Elevation Model files for the Surface.

VBA object name: AeccDEMFiles

N/A **Create using:**

Access via: Surface Inputs. DEMFiles.

Methods

Properties Add

Events Application

Delete Modified

Count

<u>Item</u>

The DescriptionKey object represents the description key for point.

VBA object name: AeccDescriptionKey

Create using: DescriptionKeyFile.Add

Access via: DescriptionKeyFile.Item

Properties

Application

Code

DescriptionFormat

DescriptionLayer

RotateByDescriptionParam

RotateByFixedFactor

RotateClockwise

RotateDescriptionParam

Methods RotateFixedFactor

Events

<u>Save</u>

Modified

<u>ScaleByDescriptionParam</u>

ScaleByDrawingScale

ScaleByFixedFactor

ScaleDescriptionParam

ScaleFixedFactor

ScaleInXY

<u>ScaleInZ</u>

SymbolBlock

SymbolLayer

Collection

The DescriptionKeyFile collection represents all of the description keys in a project description key file.

VBA object name: AeccDescriptionKeyFile

Create using: DescriptionKeyFiles.Add

Access via: DescriptionKeyFiles.Item

Properties

Methods Application

Add Count

Events

Delete FullName Modified

<u>Item</u> <u>Name</u>

Path

Collection

The DescriptionKeyFiles collection represents all of the description keys for a project description key file.

VBA object name: AeccDescriptionKeyFiles

Create using: N/A

Access via: Project.DescriptionKeyFiles

Properties

Methods

Application

<u>Add</u>

<u>AutoSave</u>

Events

<u>Delete</u>

Modified

<u>Count</u> <u>Item</u>

Path

1 ■ Document Object

An AutoCAD Land Desktop drawing.

VBA object name: AeccDocument

Create using: Documents.NewProjectBased

Access via: Documents.Item

Because AutoCAD Land Desktop works with only one document at a time, and that document represents one database, the active document can be thought of as the current database as well as the current document.

The <u>NewProjectBased</u> and <u>OpenProjectBased</u> methods handle project-based drawings specific to AutoCAD Land Desktop and override the default AutoCAD functionality.

The <u>Preferences</u> property provides access to drawing settings.

See the <u>Document</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events

provided through this object.

Properties

Application

CrossSectionBlocks (Civil Engineering Feature)

<u>MaskBlockStyles</u>

Methods

MassGroups

NewProjectBased

<u>MVBlockStyles</u>

OpenProjectBased

Preferences

ProfileBlocks (Civil Engineering Feature)

ProfileStyles

Utility

Events

None

↑ ■ Documents Collection

The collection of all AutoCAD Land Desktop drawings open in the current session.

VBA object name: AeccDocuments

Create using: N/A

Access via: Application.Documents

The documents collection will always have a single item – the currently opened drawing. To open a different drawing, use the OpenProjectBased method from the Document object.

See the <u>Documents</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.



Count

An AutoCAD Land Desktop project-based drawing.

VBA object name: AeccDrawing

Create using: N/A

Access via: Drawings.Item

The Drawing object represents an AutoCAD drawing that is associated with a AutoCAD Land Desktop project.

Properties

Application

Methods FullName

Events

None

None

Name

Path

The collection of all AutoCAD Land Desktop drawings in a project.

VBA object name: AeccDrawings

Create using: N/A

Access via: Project.Drawings

The collection of drawings at the specified drawing path. Unlike the <u>Documents</u> collection, items in the Drawings collection are not necessarily loaded into the application. Use the Drawings collection to enumerate the drawings in the project, select a Project-based drawing to open, or create a new Project-based drawing from a template

Methods Application

Item Count

Open Path

Properties

Events

Modified

ShowSubfolders

The Edge object represents a single output edge for the Surface.

VBA object name: AeccEdge

Create using: N/A

Access via: Edges.Item

Properties

Methods Application

Events

None <u>Modified</u>

Coordinates

The Edges collection represents all of the output edges for the Surface.

VBA object name: AeccEdges

Create using: N/A

Access via: SurfaceOutputs.Edges

Properties

Methods Application Events

None <u>Modified</u>

Count

↑ ■ EGProfile Object (Civil

Engineering Feature)

The EGProfile object represents the station / elevation information for a given surface.

VBA object name: AeccEGProfile

Create using: PGProfiles.Add

Access via: EGProfiles.Item

Properties

Methods Application

<u>ElevationAt</u>
<u>StationElevations</u>

Modified

Events

Import SurfaceName

<u>InstantGrade</u>

<u>Type</u>

▲ EGProfiles Collection

(Civil Engineering Feature)

The EGProfiles collection represents all of the existing ground profiles.

VBA object name: AeccEGProfiles

Create using: N/A

Access via: Alignment.EGProfiles

Methods

Add Properties

<u>Delete</u> <u>Application</u> Events

Modified

<u>Item</u> <u>Count</u>

ProfileByType

↑ ■ ElevationContour

Object

The ElevationContour object represents a single contour for the ElevationContours collection.

VBA object name: AeccElevationContour

Create using: N/A

Access via: ElevationContours.Item

 $\frac{\text{Methods}}{\text{Application}} \\ \text{None} \\ \frac{\text{Coordinates}}{\text{Coordinates}}$

↑ ■ ElevationContours

Collection

The ElevationContours collection represents all of the calculated contours at a given elevation for the Surface.

VBA object name: AeccElevationContours

N/A **Create using:**

Access via: Surface Outputs. Elevation Contours

Properties

Methods Application Events

Modified <u>Item</u> Count

Elevation

↑ Face Object

The Face object represents a single calculated face (triangle) for the Surface.

VBA object name: AeccFace

Create using: N/A

Access via: Faces.Item

The Face object represents a single face (triangle) for the current TIN Surface. The Face is a three-dimensional (planar) surface triangle.

Properties

Application

Area2D

Methods Area3D **Events**

None <u>Modified</u>

Coordinates

<u>IsVisible</u>

Normal

↑ Faces Collection

The Faces collection represents all of the calculated faces (triangles) for the Surface.

VBA object name: AeccFaces

Create using: N/A

Access via: SurfaceOutputs.Faces

Methods

FindAllFaces Properties

Application

FindFace

Count Modified

Events

FindPath

SearchType

<u>Item</u>

↑ ↓ FGProfile Object (Civil

Engineering Feature)

The FGProfile object represents a finished ground vertical PVIs.

VBA object name: AeccFGProfile

Create using: FGProfiles.Add

Access via: FGProfiles.Item

Properties

Methods Application Events

ElevationAt

PVIs Modified

<u>InstantGrade</u>

<u>Type</u>

↑ ↓ FGProfiles Collection

(Civil Engineering Feature)

The FGProfiles collection represents all of the finished grade profiles in the project.

VBA object name: AeccFGProfiles

Create using: N/A

Access via: Alignment.FGProfiles

Methods

<u>Properties</u>

Application Events

<u>Delete</u>

Count Modified

<u>Item</u>

Name

ProfileByType

↑ FileLock Object

A file lock in the AutoCAD Land Desktop project.

VBA object name: AeccFileLock

Create using: N/A

Access via: FileLocks.Item

Use this object to access properties of a particular file lock.

Properties

Application

<u>File</u>

Methods Label Events

None

Modified

LockType

Owner

<u>Time</u>

↑ ↓ FileLocks Collection

The collection of all file locks for a particular project.

VBA object name: AeccFileLocks

Create using: N/A

Access via: Project.FileLocks

The collection of all file locks for a particular project. Use the FileLocks collection to view which files in a project are locked, and who currently owns the locks. The lock owner is the person who opens the file first, locking the project files so that other people cannot make changes to them.

The delete methods are hidden since if used carelessly they can cause data corruption or loss of data. However, advanced users may require this functionality in the case of a power failure or system error.

Properties

Methods Application Events

<u>Item Count Modified</u>

<u>Path</u>

↑ Parcel Object

The Parcel object represents a parcel in the current project database.

VBA object name: AeccParcel

Create using: Parcels.Add

Access via: Parcels.Item

Properties

Application

<u>Area</u>

Methods CentroidEasting

AddCurve CentroidNorthing Events

AddLine Name Modified

<u>Import</u> <u>Number</u>

ParcelEntities

<u>Perimeter</u>

Precision

↑ ParcelCurve Object

The ParcelCurve object represents a curve entity for the parcel.

VBA object name: AeccParcelCurve

Create using: Parcel.AddCurve

Access via: ParcelEntities.Item

Properties

Application

CCWFlag

CenterEasting

CenterNorthing

ChordLength

Course

Courseln

Methods CourseOut

Events

None <u>Delta</u>

Modified

EndEasting

EndNorthing

Length

Radius

StartEasting

StartNorthing

TangentLength

Type

↑ □ ParcelEntities

Collection

The ParcelEntities collection represents all of the entities for a Parcel.

VBA object name: AeccParcelEntities

Create using: N/A

Access via: Parcel.ParcelEntities

Events

Properties

Methods **Application**

Modified <u>ltem</u>

Count

↑ ParcelEntity Collection

The ParcelEntity collection is the base class for ParcelCurve and ParcelLine.

VBA object name: AeccParcelEntity

Create using: N/A

Access via: ParcelEntities.Item

All parcel entities are derived from a ParcelEntity - they all share the methods, properties and events listed below. You can use this base class as the object type returned from the method Item of the ParcelEntities collection. If you need access to the complete details of a parcel entity, use the Type property (or TypeOf function in VB) to determine whether to assign the returned object to an ParcelLine.

Properties

Application

Course

EndEasting

Methods EndNorthing Events

None <u>Length</u> <u>Modified</u>

StartEasting

StartNorthing

Type

↑ ParcelLine Object

The ParcelLine object represents a line entity for the Parcel.

VBA object name: AeccParcelLine

Create using: Parcel.AddLine

Access via: ParcelEntities.Item

Properties

Application

Course

EndEasting

Methods EndNorthing

Events

None

Modified

Length

StartEasting

StartNorthing

<u>Type</u>

↑ Parcels Collection

The Parcels collection represents all of the parcels that are in the current project.

VBA object name: AeccParcels

Create using: N/A

Access via: Project.Parcels

Methods

Add Properties

Delete Application Events

Modified

<u>Item</u> <u>Count</u>

Rename

↑ PointFile Object

The PointFile object represents a single point file for the Surface.

VBA object name: AeccPointFile

Create using: PointFiles.Add

Access via: PointFiles.Item

Properties

Methods Application Events

None Format Modified

Name

↑ ▶ PointFiles Collection

The PointFiles collection represents all of the point files for the Surface.

VBA object name: AeccPointFiles

Create using: N/A

Access via: SurfaceInputs.PointFiles

Methods

Delete

Add **Properties**

<u>Application</u> Events

Modified

Count

<u>ltem</u>

1 ■ PointGroup Object

The PointGroup collection represents a Point Group in the project database.

VBA object name: AeccPointGroup

Create using: PointGroups.Add

Access via: PointGroups.Item

Properties

Application

Description

DescriptionOverride

DescriptionXDRef

Elevation

FlevationOverride

Methods

ElevationXDRef

ClearOverrides

Events

<u>Save</u>

GroupName

Modified

<u>LabelStyle</u>

LabelStyleOverride

LabelStyleXDRef

<u>Name</u>

NameOverride

NameXDRef

PointList

1 ■ PointGroupName

Object

The PointGroupName object represents a single PointGroupName for the PointGroupNames collection.

VBA object name: AeccPoiintGroupName

Create using: PointGroupNames.Add

Access via: PointGroupNames.Item

By using PointGroupNames, you can add Cogo Point Groups to the Surface definition. A Cogo Point Group will contain a range of Cogo Points. By using these objects, a Surface can be built using ranges of Cogo Points.

No validation exists between the PointGroupName and a Cogo Point Group. When a PointGroupName is added, AutoCAD Land Desktop does not check to see if the PointGroupName.Name actually exists as a Cogo Point Group.

Properties

Methods Application Events

None None

<u>Name</u>

→ U PointGroupNames

Collection

The PointGroupNames collection represents all of the PointGroupNames for the current Surface.

VBA object name: AeccPointGroupNames

Create using: N/A

Access via: SurfaceInputs.PointGroupNames

By using PointGroupNames, you can add Cogo Point Groups to the Surface definition. A Cogo Point Group will contain a range of Cogo Points. By using these objects, a Surface can be built using ranges of Cogo Points.

No validation exists between the PointGroupName and a Cogo Point Group. When a PointGroupName is added, AutoCAD Land Desktop does not check to see if the PointGroupName.Name actually exists as a Cogo Point Group.

Methods

Properties <u>Add</u>

<u>Application</u> Events

<u>Delete</u> **Modified**

Count

<u>Item</u>

The PointGroups collection represents all of the Point Groups in the project database.

VBA object name: AeccPointGroups

Create using: N/A

Access via: Project.PointGroups

Cogo PointGroups allow the user to work with a subset of Cogo points. This subset can also have the elevation, description, and name of the points overridden. This would be useful for when a point would need multiple elevations.

Methods Properties

Add Application Events

Delete AutoSave Modified

<u>Item</u> <u>Count</u>

↑ Preferences Object

This object specifies the current AutoCAD Land Desktop settings.

VBA object name: AeccPreferences

Create using: N/A

Access via: Application.Preferences

The Preferences object holds all the options stored external to the drawing such as User Preferences. Options that reside in the drawing can be accessed through the <u>DatabasePreferences</u> object.

The Preferences object is divided into separate objects, with each representing a set of related options.

The Preferences object can be referenced from the <u>Preferences</u> property on the Application object.

See the <u>Preferences</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Properties

Methods Application Events

None <u>Files</u>

None

<u>User</u>

Object

This object specifies the settings for Horizontal Alignments.

VBA object name: AeccPreferencesAlignment

Create using: N/A

Access via: PreferencesProject.PreferencesAlignment

Methods

<u>GetDouble</u>

GetInterger

GetString Properties Events

Application Modified

SetDouble

<u>SetInterger</u>

· · · ·

<u>SetString</u>

↑ PreferencesCogo Object

This object specifies the settings for Points.

VBA object name: AeccPreferencesCogo

Create using: N/A

Access via: PreferencesProject.PreferencesCogo

Methods

<u>GetDouble</u>

<u>GetInterger</u>

<u>GetString</u>

Properties Events

Application Modified

SetDouble

<u>SetInterger</u>

SetString

PreferencesCrossSection

Object (Civil Engineering Feature)

This object specifies the settings for Alignment Cross Sections.

VBA object name: AeccPreferencesCrossSection

Create using: N/A

Access via: PreferencesProject.PreferencesCrossSection

Methods

GetDouble

GetInteger

GetString Properties Events

Application Modified

<u>SetDouble</u>

<u>SetInteger</u>

<u>SetString</u>

↑ PreferencesFiles Object

This object specifies the paths used by the AutoCAD Land Desktop.

VBA object name: AeccPreferencesFiles

Create using: N/A

Access via: Preferences.PreferencesFiles

This object specifies the paths used by the program. Some paths are established during installation and are exposed as read-only properties.

Paths are returned without a trailing backslash delimiter "\".

Changes to <u>BorderPath</u>, <u>CivilDataFilesPath</u>, <u>ContourStylesPath</u>, <u>DrawingSetupPath</u>, <u>FormatsPath</u>, <u>LabelStylePath</u>, <u>PreferencesPath</u>, <u>ProjectPath</u>, <u>PrototypePath</u>, <u>SpeedTablesPath</u>, <u>SymbolManagerPath</u> and <u>TempPath</u> will not take affect until you exit the program and restart.

See the <u>PreferencesFiles</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Properties

Application

BorderPath

CivilDataFilesPath

ContourStylesPath

DataPath

DrawingSetupPath

FormatsPath

Methods HelpPath

Events

None <u>LabelStylePath</u>

None

PreferencesPath

ProgramPath

ProjectPath

PrototypePath

<u>SpeedTablesPath</u>

SymbolManagerPath

SystemPath

TempPath

↑ PreferencesParcel

Object

This object specifies the settings for Parcels.

VBA object name: AeccPreferencesParcel

Create using: N/A

Access via: PreferencesProject.PreferencesParcel

Methods

<u>GetInteger</u>

GetString Properties

Events

Application Modified

<u>SetInteger</u>

SetString

↑ PreferencesProfile

Object (Civil Engineering Feature)

This object specifies the settings for Alignment Vertical Profiles.

VBA object name: AeccPreferencesProfile

Create using: N/A

Access via: PreferencesProject.PreferencesProfile

Methods

GetDouble

GetInteger

GetString Properties Events

Application Modified

SetDouble

<u>SetInteger</u>

· · · ·

<u>SetString</u>

↑ PreferencesProject

Object

This object specifies the project settings.

VBA object name: AeccPreferencesProject

Create using: N/A

Access via: Project.Preferences

The PreferencesProject object holds all the options associated with external data such as Point Settings, Alignments, etc.

The PreferencesProject object can be referenced from the <u>Preferences</u> property on the <u>Project</u> object.

Properties

Application

Alignment

Methods Cogo

Events

None <u>CrossSection</u> (Civil Engineering Feature) <u>Modified</u>

<u>Parcel</u>

Profile (Civil Engineering Feature)

Surface

↑ PreferencesSurface

Object

This object specifies the settings for Surfaces.

VBA object name: AeccPreferencesSurface

Create using: N/A

Access via: PreferencesProject.PreferencesSurface

Methods

GetDouble

GetInterger

GetString Properties Events

Application Modified

SetDouble

<u>SetInterger</u>

- · - · ·

<u>SetString</u>

↑ PreferencesUser Object

This object specifies the options maintained on a per-user basis.

VBA object name: AeccPreferencesUser

Create using: N/A

Access via: Preferences.PreferencesUser

The User Preferences control program-wide preferences such as the AutoCAD overrides and the drawing setup method.

These settings are saved at the <u>PreferencesPath</u> in the file <AutoCAD login name>.dfm.

Changes to <u>OverrideNew</u>, <u>OverrideOpen</u> and <u>ShowStartupDialog</u> will not take affect until you exit the program and restart.

See the <u>PreferencesUser</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Properties

Application

<u>FirstTimeDrawingSetup</u>

<u>FirstTimeDrawingSetupFile</u>

LastUsedDwg

Methods LastUsedDwgPath

Events

None

None

LastUsedProj

LastUsedProjPath

OverrideNew

<u>OverrideOpen</u>

ShowStartupDialog

↑ ProfileBlock Object

(Civil Engineering Feature)

The ProfileBlock object represents a profile in the current drawing.

VBA object name: AeccProfileBlock

Create using: N/A

Access via: ProfileBlocks.Item

Properties

Application

Coordinates

CurveLabelIncrement

DatumElevation

Direction

Methods ____ . .

EGPrecision

<u>GetLayerName</u>

EndingStation

FGPrecision

Events

<u>StationElevationToXy</u>

Modified

XyToStationElevation

Name

StartingStation

StationIncrement

TangentLabelIncrement

UpperRight

VerticalScale

(Civil Engineering Feature)

The ProfileBlocks collection represents all of the profiles that are in the current drawing.

VBA object name: AeccProfileBlocks

Create using: N/A

Access via: Document.ProfileBlocks

 $\frac{\text{Methods}}{\text{Application}} \frac{\text{Events}}{\text{Modified}}$

Count

↑ Project Object

An AutoCAD Land Desktop project.

VBA object name: AeccProject

Create using: Projects.Add

Access via: Projects.Item Application.ActiveProjects

The <u>Project</u> object represents a project in the AutoCAD Land Desktop. Projects are used to manage and organize all the data for a job that you are working on. This data includes the project point file, alignment database, parcel database, surface database, drawing files, and so on.

AutoCAD Land Desktop works with only one project at a time. You can query certain properties of all Project objects in the <u>Projects</u> collection, but access to project-based data (such as <u>CogoPoints</u>) is only allowed in the <u>ActiveProject</u>. The ActiveProject_object has its <u>Active</u> property set to TRUE.

Properties

- ..

Methods None	Active	
	<u>Alignments</u>	
	<u>Application</u>	
	Author	
	CogoPoints Description	
	<u>DescriptionKeyFiles</u>	
	<u>DrawingPath</u>	
	<u>Drawings</u>	
	<u>FileLocks</u>	Events Modified
	<u>FullName</u>	<u></u>
	Keywords	
	Name	
	<u>Parcels</u>	
	<u>Path</u>	
	<u>PointGroups</u>	
	<u>Preferences</u>	
	<u>PrototypeName</u>	
	Surfaces	

↑ Projects Collection

The collection of all AutoCAD Land Desktop projects on the network.

VBA object name: AeccProjects

Create using: N/A

Access via: Application.Projects

Use the Projects collection to find a particular project by name, author or keywords. You can also use the Projects collection to create a new project based on a prototype.

Methods Properties

Add Application Events

Delete Count Modified

Item ProjectPath

↑ Prototype Object

An AutoCAD Land Desktop project prototype.

VBA object name: AeccPrototype

Create using: Prototypes.Copy

Access via: Prototypes.Item

AutoCAD Land Desktop uses prototypes as a convenient way for you to maintain standard settings for your drawings. Whenever a new drawing is attached to a project, its default settings are copied from this prototype. The settings are copied to each drawing so that after a drawing is created, its settings can be modified independently of any of the other drawings in the project.

Properties

Methods **Application** Events

None <u>Description Modified</u>

. .

<u>Name</u>

The collection of all AutoCAD Land Desktop project prototypes.

VBA object name: AeccPrototypes

Create using: N/A

Access via: Application.Prototypes

The collection of prototypes at the **PrototypePath**.

Methods

Copy Properties

Application

Modified

Events

<u>Delete</u> <u>Count</u>

<u>Item</u>

Engineering Feature)

The PVI object represents a Point of Vertical Interstection for a Finished Ground Profile.

VBA object name: AeccPVI

Create using: PVIs.Add

Access via: PVIs.Item

Properties

Application

Methods CurveLength

None

Modified

Events

Elevation

Station



Engineering Feature)

The PVIs collection represents all of the Point of Vertical Intesection objects in the project.

VBA object name: AeccPVIs

Create using: N/A

Access via: FGProfile.PVIs

Methods

<u>Delete</u>

<u>Add</u> Properties

Application Events

Modified

Count

<u>Item</u>

The StationEquation object represents a station equation for the alignment.

VBA object name: AeccStationEquation

Create using: StationEquations.Add

Access via: StationEquations.Item

Properties

Application

Methods StationAhead

Modified

Events

None <u>None StationBack</u>

Type

↑ ↓ StationEquations

Collection

The StationEquations collection represents all of the station equations for the Alignment.

VBA object name: AeccStationsEquations

N/A **Create using:**

Access via: A lignment. Stations Equations

Methods

<u>Add</u> **Properties**

Events Delete Application

Modified

<u>ltem</u> Count

RemoveAll

(Civil Engineering Feature)

The Superelevation object represents a single superelevation for an alignment.

VBA object name: AeccSuperelevation

Create using: N/A

Access via: Superelevations.Item

Properties

Application

Events

Methods CurveCode Modified None

Station

SuperelevationCode

The Superelevations collection represents all of the superelevations for an alignment.

/BA object name:	AeccSuperelevations
Create using:	N/A
Access via:	Alignment.Superelevations
Methods	
<u>tem</u>	
Properties	

Application

Count

Events

Modified

↑ Surface Object

The Surface object represents a Surface in the project database.

VBA object name: AeccSurface

Create using: Surfaces.Add

Access via: Surfaces.Item

Properties

Application

Area2D

Area3D

<u>AverageGrade</u>

Description

Methods Inputs

<u>AddToAllElevations</u> <u>LockType</u>

<u>AreaVolume</u> <u>MaxElevation</u>

<u>Build</u> <u>MaxFaceArea</u>

FindAllConnectingEdges MaxGrade

<u>FindConnectingEdge</u> <u>MeanElevation</u>

FindPoint MinElevation Modified

<u>GetBoundingBox</u> <u>MinFaceArea</u>

<u>GetElevation</u> <u>MinGrade</u>

<u>Import</u> <u>Name</u>

<u>Paste</u> <u>NumberOfFaces</u>

<u>SampleElevations</u> <u>NumberOfPoints</u>

<u>SetBoundingBox</u> <u>Outputs</u>

PointOnLineTolerance

Events

PointTolerance

RevisionNumber

Status

<u>Type</u>

Volume

↑ ↓ SurfaceInputs

Collection

The SurfaceInputs object represents all inputs for a Surface.

VBA object name: AeccSurfaceInputs

Create using: N/A

Access via: Surface.SurfaceInputs

Properties

Application

Boundaries

Methods BreakLines

Events

None <u>ContourItems</u> <u>Modified</u>

DEMFiles

- - ---

PointFiles

<u>PointGroupNames</u>

1 ■ SurfaceOutputs Object

The SurfaceOutputs object represents all outputs for a Surface.

VBA object name: AeccSurfaceOutputs

Create using: N/A

Access via: Surface.SurfaceOutputs

Properties

Application

Edges

Events

Methods ElevationContours None

Modified

Faces

TinPoints

WaterSheds

↑ Surfaces Collection

The Surfaces collection represents all of the Surface objects in the project database.

Events

VBA object name: AeccSurfaces

Create using: N/A

Access via: Project.Surfaces

Methods

Add

Composite Properties

<u>Copy</u> <u>Application</u>

Delete Count Modified

<u>DifferenceGrid</u> <u>CurrentSurface</u>

Item

<u>.....</u>

Rename

↑ TinPoint Object

The TinPoint represents a calculated point for the Surface.

VBA object name: AeccTinPoint

Create using: N/A

Access via: TinPoints.Item

Properties

Application

Methods

Easting

Events

None

Modified

Elevation

Northing

↑ TinPoints Collection

The TinPoints represents all of the calculated points for the Surface.

VBA object name: AeccTinPoints

Create using: N/A

Access via: SurfaceOutputs.TinPoints

Properties

Methods Application Events

<u>Item</u> <u>Modified</u>

Count

↑ Utility Object

A series of methods provided for utility purposes.

VBA object name: AeccUtility

Create using: N/A

Access via: Document.Utility

See the <u>Utility</u> object in the AutoCAD ActiveX and VBA Reference for information regarding additional Methods, Properties and Events provided through this object.

Methods

EastNorthToXy Properties Events

None None

XyToEastNorth

The WaterShed object represents a single watershed item for the WaterSheds collection.

VBA object name: AeccWaterShed

N/A **Create using:**

Access via: WaterSheds.Item

Properties

Application

Boundary

Methods <u>DrainsInto</u> Events

None

Modified

<u>Id</u>

OverflowPoints

Type

1 WaterSheds Collection

The WaterSheds collection represents all of the WaterShed items for the Surface.

VBA object name: AeccWaterSheds

N/A **Create using:**

Access via: Surface Outputs. Water Sheds

Properties

Application

Methods Count

Events

Build

ExceedBoth Modified

<u>Item</u>

<u>MinDepressionArea</u>

MinDepressionDepth

A Example

```
Sub Example_A()
```

```
'This example returns the A value for the first Spiral found in the
  ' first Alignment in the collection.
  Dim align As AeccAlignment
  Dim alignEnts As AeccAlignEntity
  Set align = AeccApplication.ActiveProject.Alignments.Item(0)
  Dim alignMsg As String
  alignMsg = "There is no Spiral entity in the first Alignment."
  ' Find first Spiral in the alignment
  For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
       "The A value for the first Spiral in the alignment is: " & alignEnt.A
       Exit For
    End If
  Next
  MsgBox alignMsg, vbInformation, "A Example"
End Sub
```

Active Example

Sub Example_Active()

'This example returns the Active setting for the first Project

' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Active value for the first Project in the collection is: " & proj.Ac vbInformation, "Active Example"

ActiveDocument Example

Sub Example_ActiveDocument()

' This example returns name of the CurrentDocument.
Dim activeDoc As AeccDocument
Set activeDoc = AeccApplication.ActiveDocument

MsgBox "The ActiveDocument is: " & activeDoc.Name, _ vbInformation, "ActiveDocument Example"

ActiveProject Example

Sub Example_ActiveProject()

' This example returns the name of the ActivProject.

Dim activeProj As AeccProject

Set activeProj = AeccApplication.ActiveProject

MsgBox "The ActiveProject is: " & activeProj.Name, _ vbInformation, "ActiveProject Example"

AD1 Example

```
Sub Example_AD1()
```

```
'This example returns the AD1 value for the first Spiral found in the
  ' first Alignment in the collection.
  Dim align As AeccAlignment
  Dim alignEnts As AeccAlignEntity
  Set align = AeccApplication.ActiveProject.Alignments.Item(0)
  Dim alignMsg As String
  alignMsg = "There is no Spiral entity in the first Alignment."
  ' Find first Spiral in the alignment
  For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
       alignMsg; = "The AD1 value for the first Spiral in the alignment is: " & al
       Exit For
    End If
  Next
  MsgBox alignMsg, vbInformation, "AD1 Example"
End Sub
```

AD2 Example

```
Sub Example_AD2()
```

```
'This example returns the AD2 value for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg; = "The AD2 value for the first Spiral in the alignment is: " & al Exit For

End If

Next

MsgBox alignMsg, vbInformation, "AD2 Example"
```

Add Example

Examples:

- l Alignments
- 1 **Boundaries**
- l BreakLines
- l CogoPoints
- l ContourItems
- 1 **DEMFiles**
- 1 DescriptionKeyFile
- 1 DescriptionKeyFiles
- l <u>EGProfiles</u> (Civil Engineering Feature)
- 1 <u>FGProfiles</u> (Civil Engineering Feature)
- 1 Parcels
- l <u>PointFiles</u>
- l <u>PointGroupNames</u>
- l <u>PointGroups</u>
- l Projects
- l PVIs (Civil Engineering Feature)
- 1 StationEquations
- l Surfaces

Sub Example_Add_Alignments()

- 'This example adds an Alignment made up of a tangent,
- ' curve, and spiral entities.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)

' Add a tangent, curve, and spiral to the alignment

Dim tangent As AeccAlignTangent

Dim curve As AeccAlignCurve

Dim spiral As AeccAlignSpiral

Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

Set spiral = align.AddSpiral(150#, 0#, 388.069176379758, -5.83082052087824

Set curve = align.AddCurve(250, 100#, 320.966940499621, 197.98347024973

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti vbInformation, "Add Example"

End Sub

Sub Example_Add_Boundaries()

- 'This example starts by displays the initial count of the Boundaries
- ' for the first surface in the collection. A new visible Boundary is added
- ' and the count is redisplays. Finally, the new Boundary is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim bounds As AeccBoundaries

Dim Bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bounds = surf.Inputs.Boundaries

MsgBox "The Boundaries Count is: " & bounds.count, vbInformation, "Add E:

```
' Initialize variables
Dim pnts(0 To 14) As Double
Dim count As Integer
Dim index As Integer
Dim pnt As Variant
index = 0
```

' Add a new Boundary based on five selected points

```
For count = 1 To 5

pnt = ThisDrawing.Utility.GetPoint(, "Select point" + Str(count) + " of Bour pnts(index) = pnt(0): pnts(index + 1) = pnt(1): pnts(index + 2) = pnt(2) index = index + 3
```

Next count

' Add a new Boundary

Set Bound = bounds.Add(kBoundaryTypeShow, False, pnts, "NewBoundary")

MsgBox "The BreakLines Count after the add is: " & bounds.count, vbInforma

' Delete the new Boundary bounds.Delete Bound.Id

MsgBox "The Boundaries Count after the delete is " & bounds.count, vbInforn

End Sub

Sub Example_Add_BreakLines()

- 'This example starts by displays the initial count of the BreakLines
- ' for the first surface in the collection. A new BreakLines is added
- ' and the count is redisplays. Finally, the new file is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim brkLines As AeccBreakLines

```
Dim brkLine As AeccBreakLine
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set brkLines = surf.Inputs.BreakLines
```

MsgBox "The BreakLines Count is: " & brkLines.count, vbInformation, "Add

```
' Initialize variables
Dim pnts(0 To 14) As Double
Dim count As Integer
Dim index As Integer
Dim pnt As Variant
```

index = 0

' Add a new BreakLine based on five selected points

```
For count = 1 To 5

pnt = ThisDrawing.Utility.GetPoint(, "Select point" + Str(count) + " of Breal pnts(index) = pnt(0): pnts(index + 1) = pnt(1): pnts(index + 2) = pnt(2) index = index + 3
```

Next count

Set brkLine = brkLines.Add(pnts, "NewBreakLine")

MsgBox "The BreakLines Count after the add is: " & brkLines.count, vbInforn

' Delete the new BreakLines brkLines.Delete brkLine.Id

MsgBox "The BreakLines Count after the delete is " & brkLines.count, vbInfo

End Sub

Sub Example_Add_CogoPoints()

'This function adds a new CogoPoint to the CogoPoints collection.

Dim cogoPnts As AeccCogoPoints

Dim newCogoPnt As AeccCogoPoint

```
Set cogoPnts = AeccApplication.ActiveProject.CogoPoints
  'Get a point
  Dim newPnt As Variant
  newPnt = ThisDrawing.Utility.GetPoint(, "Select point location: ")
  ' Add CogoPoint
  Set newCogoPnt = cogoPnts.Add(newPnt, kCoordinateFormatXYZ)
  MsgBox "The Number for the new CogoPoint is: " & newCogoPnt.Number, vl
End Sub
Sub Example_Add_ContourItems()
  'This example Adds a Contour, with an elevation of 100,
  ' to the first collection in ContourItems.
  Dim surf As AeccSurface
  Dim conts As AeccContourItems
  Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Set conts = surf.Inputs.ContourItems
  ' Display the number of ContourItems in the collection before the add
  MsgBox "The number of ContourItems in the collection is: " & conts.count, vt
  'Initialize variables
  Dim pnts(0 To 29) As Double
  Dim count As Integer
  Dim index As Integer
  Dim pnt As Variant
  index = 0
  'Define a new contour based upon 10 entered points
  For count = 1 \text{ To } 10
    pnt = ThisDrawing.Utility.GetPoint(, "Select point" + str(count) + " of Conto
    pnts(index) = pnt(0): pnts(index + 1) = pnt(1): pnts(index + 2) = 100#
```

index = index + 3
Next count

' Add Contour to the collection conts.Add Points

' Display the number of ContourItems in the collection after the add MsgBox "The number of ContourItems in the collection is: " & conts.count, vt

End Sub

Sub Example_Add_DEMFiles()

- 'This example starts by displays the initial count of DEMFiles
- ' for the first surface. A new DEMfile is added and the count is redisplays.
- ' Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim surf As AeccSurface

Dim DEMFiles As AeccDEMFiles

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set DEMFiles = surf.Inputs.DEMFiles

MsgBox "The DEMFiles Count is: " & DEMFiles.Count, vbInformation, "Add

'Get the DEMFile name and format the prompt

Dim DEMName As String

Dim prompt As String

prompt = "Enter the name of the DEMFile for surface " & surf.Name & ": "
DEMName = ThisDrawing.Utility.GetString(False, prompt)

' Add a new DEMFile DEMFiles.Add DEMName

MsgBox "The DEMFiles Count after the add is: " & DEMFiles.Count, vbInfor

^{&#}x27; Delete the new DEMFile

DEMFiles.Delete DEMName

MsgBox "The DEMFiles Count after the delete is " & DEMFiles.Count, vbInfo

End Sub

Sub Example_Add_DescriptionKeyFile()

- 'This example starts by displays the initial count of DescriptionKeys
- ' in the DEFAULT DescriptionKey file. A new DescriptionKey is added
- ' and the count is redisplays. Finally, the new key is deleted. The
- ' count is displayed again, showing the deletion.

Dim dkeyFile As AeccDescriptionKeyFile

Set dkeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item("DEI

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "A

' Add a new DescriptionKey dkeyFile.Add "New"

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "A

' Delete the new DescriptionKey dkeyFile.Delete "New"

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "A

End Sub

;Sub Example_Add_DescriptionKeyFiles()

- 'This example starts by displays the initial count of DescriptionKeyFiles
- ' on the system. A new DescriptionKeyFile is added and the count is redisplays
- ' Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim dKeyFiles As AeccDescriptionKeyFiles

Set dKeyFiles = AeccApplication.ActiveProject.DescriptionKeyFiles

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

' Add a new DescriptionKeyFile dKeyFiles.Add "NewFile"

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

' Delete the new DescriptionKeyFile dKeyFiles.Delete "NewFile"

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

End Sub

Sub Example_Add_EGProfiles()

'This example gets the surface name from the user and adds a new

'existing ground profile for the first alignment in the collection.

Dim align As AeccAlignment

Dim newEGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

'Get the surface name and format the prompt

Dim surfName As String

Dim prompt As String

prompt = "Enter the name of the surface for alignment " & align.Name & ": "
surfName = ThisDrawing.Utility.GetString(False, prompt)

'Add the new existing ground profile
Set newEGProf = align.EGProfiles.Add(kEgCenter, surfName)

MsgBox "The first station for the new existing ground profile is: "_ & newEGProf.StationElevations(0), vbInformation, "Add Example"

Sub Example_Add_FGProfiles()

'This example adds a center type finished ground profile

Dim align As AeccAlignment

Dim newFGProf As AeccFGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

'Add the new finished ground profile Set newFGProf = align.FGProfiles.Add(kFgCenter)

MsgBox "The type for the new existing ground profile is: " _ & newFGProf.Type, vbInformation, "Add Example"

End Sub

Sub Example_Add_Parcels()

- 'This example starts by displays the initial count of Parcels
- ' A new Parcel is added and the count is redisplayed. Finally,
- ' the new Parcel is deleted. The count is displayed again, showing
- ' the deletion.

Dim parcels As AeccParcels
Set parcels = AeccApplication.ActiveProject.Parcels

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

' Add a new Parcel parcels. Add "NewParcel"

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

' Delete the new Parcel parcels.Delete "NewParcel"

^{&#}x27; to the first alignment in the collection.

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

End Sub

Sub Example_Add_PointFiles()

- 'This example starts by displays the initial count of PointFiles
- ' on the system. A new Pointfile is added and the count is redisplays.
- 'Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim surf As AeccSurface

Dim pntFiles As AeccPointFiles

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set pntFiles = surf.Inputs.PointFiles

MsgBox "The PointFiles Count is: " & pntFiles.Count, vbInformation, "Add E

' Add a new PointFile pntFiles.Add "NewPointFile"

MsgBox "The PointFiles Count after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add its point after the add

' Delete the new PointFile pntFiles.Delete "NewPointFile"

MsgBox "The PointFiles Count after the delete is " & pntFiles.Count, vbInforn

End Sub

Sub Example_Add_PointGroupNames()

- 'This example starts by displays the initial count of the PointGroupNames
- ' for the first surface in the collection. A new PointGroupName is added
- ' and the count is redisplays. Finally, the new file is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim pntGrpNames As AeccPointGroupNames

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set pntGrpNames = surf.Inputs.PointGroupNames

MsgBox "The PointGroupNames Count is: " & pntGrpNames.Count, vbInform

' Add a new PointGroupName pntGrpNames.Add "NewPointGroupName"

MsgBox "The PointGroupNames Count after the add is: " & pntGrpNames.Co

' Delete the new PointGroupName pntGrpNames.Delete "NewPointGroupName"

MsgBox "The PointGroupNames Count after the delete is " & pntGrpNames.C

End Sub

Sub Example_Add_PointGroups()

'This function adds a new PointGroup named "New Group".

Dim pntGrps As AeccPointGroups

Dim newPntGrp As AeccPointGroup

Set pntGrps = AeccApplication.ActiveProject.PointGroups

' Show the number of alignments in the project MsgBox "The initial Count of PointGroups is: " & pntGrps.Count, vbInformati

' Add PointGroup
Set newPntGrp = pntGrps.Add("Example Group", "1-10, 20-30")

' Show the number of alignments in the project after the Add MsgBox "The Count of PointGroups is: " & pntGrps.Count, vbInformation, "A

End Sub

Sub Example_Add_Projects()

'This example creates a new project.

Dim projs As AeccProjects

Set projs = AeccApplication.Projects

' Show the number of projects

MsgBox "The initial Count of Projects is: " & projs.Count, vbInformation, "Ac

projs.Add "New Project", "Default (Feet)"

' Show the number of projects after the Add

MsgBox "The Count of Projects is: " & projs.Count, vbInformation, "Add Exa

End Sub

Sub Example_Add_PVIs()

'This example adds a PVI in the first finished

'ground profile of the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

MsgBox "The number of PVIs in the finished ground profiles is: " _ & FGProf.PVIs.Count, vbInformation, "Add Example"

'Get the station, elevation, and curve length

Dim station As Double

Dim elevation As Double

Dim curvelength As Double

station = ThisDrawing.Utility.GetReal("Enter a station on finished grade profil elevation = ThisDrawing.Utility.GetReal("Enter an elevation on finished grade station = ThisDrawing.Utility.GetReal("Enter a curve length on finished grade

'Add the PVI

FGProf.PVIs.Add station, elevation, curvelength

MsgBox "The number of PVIs in the finished ground profiles is: " _ & FGProf.PVIs.Count, vbInformation, "Add Example"

End Sub

Sub Example_Add_StationEquations()

- 'This example adds a StationEquation to
- ' the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Get the first alignment Set align = aligns.Item(0)

' Set the first alignment current. aligns.CurrentAlignment = align.Name

Dim staEqus As AeccStationEquations Dim staEqu As AeccStationEquation Set staEqus = align.StationEquations

' Show the StationEquation count for the first alignment MsgBox "The StationEquation count for the first alignment is: " & staEqus.cou vbInformation, "Add Example"

' Add a new StationEquations
Set staEqu = staEqus.Add(100, 50, kIncreasing)

' Show the stationEquation count after the add

MsgBox "The StationEquation count for the first alignment is: " & staEqus.cou vbInformation, "Add Example"

'This example creates a new Surface and adds it to the Surfaces collection

Dim surfs As AeccSurfaces
Set surfs = AeccApplication.ActiveProject.Surfaces

Surfs.Add "New Surface"

MsgBox "The Number of Surfaces has increased to " & surfs.Count, vbInform

AddCurve Example

Examples:

1 Alignment

1 Parcel

Sub Example_AddCurve_Alignment()

- 'This example adds an Alignment made up of a tangent,
- ' curve, and spiral entities.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)

' Add a tangent, curve, and spiral to the alignment

Dim tangent As AeccAlignTangent

Dim curve As AeccAlignCurve

Dim spiral As AeccAlignSpiral

Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

Set spiral = align.AddSpiral(150#, 0#, 388.069176379758, -5.83082052087824 Set curve = align.AddCurve(250, 100#, 320.966940499621, 197.98347024973

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti

End Sub

Sub Example_AddCurve_Parcel()

- 'This example adds a Parcel made up of lines and an arc.
- 'The parcel is then imported into the drawing.

Dim parcels As AeccParcels
Dim parcel As AeccParcel
Set parcels = AeccApplication.ActiveProject.parcels

' Add a new Parcel
Set parcel = parcels.Add("New Parcel")

' Add lines and a curve to the Parcel parcel.AddLine 50#, 50#, 150#, 50# parcel.AddLine 150#, 50#, 150#, 200# parcel.AddCurve 150#, 200#, 100#, 200#, 50#, 200#, True parcel.AddLine 50#, 200#, 50#, 50#

' Import the new Parcel parcel.Import

MsgBox "The total number of entities in the parcel is: " & parcel.ParcelEntities vbInformation, "AddCurve Example"

AddLabel Example

Sub Example_AddLabelAt()

```
'This example adds a label to a Contour object by selecting the
```

On Error Resume Next

```
' Delete existing SelectionSet
ThisDrawing.SelectionSets("SSet").Delete
```

```
'Create the selection set based on a point selection
```

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim mode As Integer Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AECC_CONTOUR"

Dim groupCode As Variant Dim dataCode As Variant Dim returnPnt As Variant

groupCode = gpCode
dataCode = dataValue
returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point on a contour line: ")
ssetObj.SelectAtPoint returnPnt, groupCode, dataCode

Dim Ent As AeccContour
Set Ent = ssetObj.Item(0)
Ent.AddLabelAt returnPnt(0), returnPnt(1)

^{&#}x27; contour

^{&#}x27; and filter for Contour objects

AddLine Example

Sub Example_AddLine()

- 'This example adds a Parcel made up of lines and an arc.
- 'The parcel is then imported into the drawing.

Dim parcels As AeccParcels

Dim parcel As AeccParcel

Set parcels = AeccApplication.ActiveProject.parcels

' Add a new Parcel
Set parcel = parcels.Add("New Parcel")

' Add lines and a curve to the Parcel parcel.AddLine 50#, 50#, 150#, 50# parcel.AddLine 150#, 50#, 150#, 200# parcel.AddCurve 150#, 200#, 100#, 200#, 50#, 200#, True parcel.AddLine 50#, 200#, 50#, 50#

' Import the new Parcel parcel.Import

MsgBox "The total number of entities in the parcel is: " & parcel.ParcelEntities vbInformation, "AddLine Example"

AddSpiral Example

Sub Example_AddSpiral()

'This example adds an Alignment made up of a tangent,

' curve, and spiral entities.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)

' Add a tangent, curve, and spiral to the alignment

Dim tangent As AeccAlignTangent

Dim curve As AeccAlignCurve

Dim spiral As AeccAlignSpiral

Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

Set spiral = align.AddSpiral(150#, 0#, 388.069176379758, -5.83082052087824

Set curve = align.AddCurve(250, 100#, 320.966940499621, 197.98347024973

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti

AddTangent Example

Sub Example_AddTangent()

'This example adds an Alignment made up of a tangent,

' curve, and spiral entities.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)

' Add a tangent, curve, and spiral to the alignment

Dim tangent As AeccAlignTangent

Dim curve As AeccAlignCurve

Dim spiral As AeccAlignSpiral

Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

Set spiral = align.AddSpiral(150#, 0#, 388.069176379758, -5.83082052087824

Set curve = align.AddCurve(250, 100#, 320.966940499621, 197.98347024973

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti

AddToAllElevations Example

Sub Example_AddToAllElevations()

'This example uses AddToAllElevations to increase the elevation by 20.0 for

' the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MeanElevation for the first surface before elevation increase is: surf.MeanElevation;, vbInformation, "AddToAllElevations Example"

surf.AddToAllElevations 20# surf.Build

MsgBox "The MeanElevation for the first surface after elevation increase is: " surf.MeanElevation;, vbInformation, "AddToAllElevations Example"

AlignEntities Example

```
Sub Example_AlignEntities()
```

```
'This example gets the count of AlignEntities for the
' first alignment in the collection.
Dim align As AeccAlignment
Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Dim count As Integer
' Get the number of entities
count = align.AlignEntities.count
If count = 0 Then
  MsgBox; "The first Alignment in the collection has no entities.", _
    vbInformation;, "AlignEntities Example"
ElseIf count = 1 Then
  MsgBox; "The first Alignment in the collection has 1 entity.", _
    vbInformation;, "AlignEntities Example"
ElseIf count > 1 Then
  MsgBox; "The firstAlignment in the collection has " & count & " entities.",
    vbInformation;, "AlignEntities Example"
End If
```

Alignment Example

Sub Example_Alignment()

'This example returns the PreferencesAlignment object that

' is used to access the StationLayer property

Dim alignPref As AeccPreferencesAlignment

Set alignPref = AeccApplication.ActiveProject.preferences.Alignment

MsgBox "The Alignment preferences for StationLayer is: " & alignPref.GetStr

AlignmentFromObjectID Example

Sub Example_AlignmentFromObjectID()

```
'This example returns the AlignmentFromObjectID for a selected alignment.
Dim aligns As AeccAlignments
Set aligns = AeccApplication.ActiveProject.Alignments
Dim alignObj As AcadObject
Dim basePnt As Variant
Dim alignID As Long
Dim alignNames As Variant
On Error Resume Next
'The following example prompts for a selection from the user
RETRY;:
ThisDrawing.Utility.GetEntity alignObj, basePnt, "Select an object"
If Err <> 0 Then
  Err.Clear
  MsgBox "Good Bye.", vbInformation, "AlignmentFromObjectID Example"
  Exit Sub
Else
  alignID = alignObj.ObjectID
  alignNames = aligns.AlignmentFromObjectID(alignID)
  MsgBox "The selected alignment is named: " & alignNames(0), _
    vbInformation, "AlignmentFromObjectID Example"
End If
GoTo RETRY
```

Alignments Example

Sub Example_Alignments()

'This example returns the tick increment for Alignments Preferences

' in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The tick increment for Alignments Preferences in the current Project , vbInformation, "Alignments Example"

Angular Azimuth Example

Sub Example_AngularAzimuth()

```
'This example returns the AngularAzimuth setting for the current drawing.
Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.Preferences
'Convert the constant to a string.
Dim Style As String
Dim AngAzm As Long
' Determine Angular Azimuth setting
AngAzm = dbPref.AngularAzimuth
Select Case AngAzm
Case 0
  Style = "bearings."
Case 1
  Style = "north azimuths."
Case 2
  Style = "south azimuths."
End Select
MsgBox "The current value for AngularAzimuth is " & Style, vbInformation, '
```

Angular Display Format Example

Sub Example_AngularDisplayFormat()

```
'This example returns the AngularDisplayFormat setting for the current drawir Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

'Convert the constant to a string.

Dim strUnits As String

If dbPref.AngularDisplayFormat = 1 Then
    strUnits = "degrees."

Else
    strUnits = "grads."

End If

MsgBox "The current value for AngularDisplayFormat is " & strUnits, _
```

vbInformation, "AngularDisplayFormat Example"

AngularPrecision Example

Sub Example_AngularPrecision()

' This example returns the AngularPrecision setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for AngularPrecision is " & dbPref.AngularPrecisi vbInformation, "AngularPrecision Example"

Application Example

Sub Example_Application()

- 'This example uses the AeccApplication property to navigate to the top of
- ' the object hierarchy to compare the current project with the last used project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

```
MsgBox "The current project is " & proj.Application.ActiveProject.Name & _ " and the last used project is " & _ proj.Application.Preferences.User.LastUsedProj, _ vbInformation, "Application Example"
```

Area Example

Sub Example_Area()

'This example returns the Area for the first parcel in the collection.

Dim parcel As AeccParcel

Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

MsgBox "The Area for the first Parcel in the collection is: "_ & Format(parcel.Area, "0.00"), vbInformation, "Area Example"

Area2D Example

Examples:

1 Face

1 Surface

Sub Example_Area2D_Face()

- 'This example returns the Area2D value for the
- ' first Face in the collection.

Dim face As AeccFace

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set face = surf.Outputs.Faces.Item(0)

MsgBox "The Area2D value is " & face. Area2D, vbInformation, "Area2D Exa

End Sub

Sub Example_Area2D_Surface()

'This example returns the Area2D for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The Area2d for the first surface is: " & surf.Area2D, vbInformation,

Area3D Example

Examples:

1 Face

1 Surface

Sub Example_Area3D_Face()

- 'This example returns the Area3D value for the
- ' first Face in the collection.

Dim face As AeccFace

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set face = surf.Outputs.Faces.Item(0)

MsgBox "The Area3D value is " & face. Area3D, vbInformation, "Area3D Exa

End Sub

Sub Example_Area3D_Surface()

'This example returns the Area3D for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The Area3D for the first surface is " & surf.Area3D, vbInformation,

AreaDisplayUnit Example

Sub Example_AreaDisplayUnit()

```
'This example returns the AreaDisplayUnit setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

'Convert the area display unit to a string.

Dim unit As String

Select Case dbPref.AreaDisplayUnit
```

Select Case dbPref.AreaDisplayUnit
Case aecUnitSquareInch
unit = "square inch."
Case aecUnitSquareFoot
unit = "square foot."
Case aecUnitSquareYard
unit = "square yard."
Case aecUnitSquareMil
unit = "square millimeters."
Case aecUnitSquareCentimeters
unit = "square centimeters
unit = "square decimeters."
Case aecUnitSquareDecimeters
unit = "square decimeters."
Case aecUnitSquareMeters
unit = "square meters."
End Select

MsgBox "The current value for AreaDisplayUnit is " & unit, vbInformation, "A

AreaPrecision Example

Sub Example_AreaPrecision()

' This example displays the AreaPrecision setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The current value for AreaPrecision is: " & dbPref.AreaPrecision, vt

AreaSuffix Example

Sub Example_AreaSuffix()

'This example displays the AreaSuffix setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The current value for AreaSuffix is: " & dbPref.AreaSuffix, _ vbInformation, "AreaSuffix Example"

AreaVolume Example

```
Sub Example_AreaVolume()
```

```
'This example returns the Cut, Fill and Net Volumes of the first surface in the
Dim surf As AeccSurface
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Dim count as Long
Dim index as Long
Dim cut As Double
Dim fill As Double
Dim tolerance As Double
Dim net As Double
Dim pnt as Variant
Dim areaPnts(0 To 14) As Double
index = 0
'Define a area based upon 5 entered points
For count = 1 To 5
  pnt = ThisDrawing.Utility.GetPoint(, "Select point" + str(count) + " for area:
  areaPnts(index) = pnt(0): areaPnts(index + 1) = pnt(1): areaPnts(index + 2) = pnt(1)
  index = index + 3
Next count
'Get volume data
surf.AreaVolume 1.0, areaPnts, cut, fill, net
MsgBox "The AreaVolumn data for the first surface is:" & vbCrLf &
  " Cut: " & cut & vbCrLf & _
  " Fill: " & fill & vbCrLf &
  " Net: " & net, vbInformation, "AreaVolumn Example"
```

ArrayToPointString Example

Sub Example_ArrayToPointString()

```
'This examples returns a string of point numbers in CogoPoints format. Dim cogoPnts As AeccCogoPoints

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

Dim pntString As String
Dim pntArray(0 To 8) As Long

'Create an array with the point numbers 1,4,5,6,8,40,10,7,32
pntArray(0) = 1: pntArray(1) = 4: pntArray(2) = 5
pntArray(3) = 6: pntArray(4) = 8: pntArray(5) = 40
pntArray(6) = 10: pntArray(7) = 7: pntArray(8) = 32

'Convert array to string
pntString = cogoPnts.ArrayToPointString(pntArray)

MsgBox "The CogoPoints point number string is " & pntString, __ vbInformation, "ArrayToPointString Example"
```

Author Example

Sub Example_Author()

- 'This example returns the Author setting for the first Project
- ' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Author value for the first Project in the collection is: " & proj.A vbInformation, "Author Example"

AutoSave Example

Examples:

- 1 Alignments
- l CogoPoints
- 1 <u>DescriptionKeyFiles</u>
- l PointGroups

Sub Example_AutoSave_Alignments()

- 'This example changes the Description and StartingStation
- ' for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = Aligns.Item(0)

- ' Make the first alignment in the collection current aligns.CurrentAlignment = align.Name
- ' Set AutoSave to FALSE to prevent writing the changes
- ' until they are all made

aligns.AutoSave = False

- ' Make changes align.Description = "New Description" align.StartingStation = 20#
- ' Save changes align.Save
- 'Reset AutoSave to the default state of TRUE aligns.AutoSave = True

MsgBox "The first Alignments Description is: " & align.Description & vbCrLf "The first Alignments StartingStation is: " & align.StartingStation, _ vbInformation, "AutoSave Example"

End Sub

```
Sub Example_AutoSave_CogoPoints()
```

```
'This example changes the Northing, Easting, and Elevation
```

' for the first CogoPoint in the collection.

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

Set cogoPnt = cogoPnts.Item(0)

cogoPnts.AutoSave = False

' Make changes

cogoPnt.Northing = 100#

cogoPnt.Easting = 100#

cogoPnt.Elevation = 150#

' Save changes

cogoPnt.Save

'Reset AutoSave to the default state of TRUE cogoPnts.AutoSave = True

MsgBox "The following changes were made to CogoPoint " & cogoPnt.Numb

- " Northing: " & cogoPnt.Northing & vbCrLf & _
- " Easting: " & cogoPnt.Easting & vbCrLf & _
- " Elevation: " & cogoPnt.Elevation, vbInformation, "AutoSave Example"

^{&#}x27; Set AutoSave to FALSE to prevent writing the changes

^{&#}x27; until they are all made

Sub Example_AutoSave_DescriptionKeyFiles()

```
'This example changes the layer settings
```

Dim dKeyFiles As AeccDescriptionKeyFiles

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFiles = AeccApplication.ActiveProject.DescriptionKeyFiles

Set dKeyFile = dKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

dKeyFiles.AutoSave = False

' Make changes

dKey.DescriptionLayer = "dKey Desc"

dKey.SymbolLayer = "dKey Symb"

' Save changes

dKey.Save

'Reset AutoSave to the default state of TRUE

dKeyFiles.AutoSave = True

MsgBox "The DescriptionLayer is: " & dkey.DescriptionLayer & vbCrLf & _ "The SymbolLayer is: " & dkey.SymbolLayer, vbInformation, "AutoSave Ex

End Sub

Sub Example_AutoSave_PointGroups()

Dim pntGrps As AeccPointGroups

Dim pntGrp As AeccPointGroup

^{&#}x27; for the first DescriptionKey in the collection.

^{&#}x27; Set AutoSave to FALSE to prevent writing the changes

^{&#}x27; until they are all made

^{&#}x27;This example changes the Description and Elevation

^{&#}x27; for the first PointGroup in the collection.

Set pntGrps = AeccApplication.ActiveProject.PointGroups
Set pntGrp = pntGrps.Item(0)

- 'Set AutoSave to FALSE to prevent writing the changes
- ' until they are all made pntGrps.AutoSave = False
- ' Make changes pntGrp.Description = "New Description" pntGrp.Elevation = 100#
- ' Save changes pntGrp.Save
- 'Reset AutoSave to the default state of TRUE pntGrps.AutoSave = True

MsgBox "The first PointGroup Description is: " & pntGrp.Description & vbCr "The first PointGroup Elevation is: " & pntGrp.Elevation, vbInformation, "A

AverageGrade Example

Sub Example_AverageGrade()

' This example returns the Average Grade for the first surface in the collection. Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The AverageGrade for the first surface is: " & Format(surf.AverageGrade vbInformation, "AverageGrade Example"

BasePoint Example

```
Sub Example_BasePoint()
```

```
'This example displays the BasePoint setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

Dim varPoint As Variant

Dim point(0 To 2) As Double

Dim basePoint As String

'Get the base point

varPoint = dbPref.BasePoint

point(0) = varPoint(0)

point(1) = varPoint(1)

point(2) = varPoint(2)

'Format base point to a string

basePoint = Format(point(0)) + ", " + Format(point(1)) + ", " + Format(point(2))

MsgBox "The current value for BasePoint is: " & basePoint, vbInformation, "E
```

BasePointNE Example

```
Sub Example_BasePointNE()
```

```
'This example displays the BasePointNE setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

Dim varPoint As Variant

Dim point(0 To 2) As Double

Dim basePointNE As String

'Get the base point

varPoint = dbPref.basePointNE

point(0) = varpoint(0)

point(1) = varpoint(1)

point(2) = varpoint(2)

'Format base point to a string

basePointNE = Format(point(0)) + ", " + Format(point(1)) + ", " + Format(1) + Format(1
```

BeginCondition Example

Sub Example_BeginCondition()

```
'This example returns the BeginCondition for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The BeginCondition for the first Spiral in the alignment is: "
Exit For

End If

Next

MsgBox alignMsg, vbInformation, "BeginCondition Example"
```

BorderBlockFilename Example

Sub Example_BorderBlockFilename()

' This example returns the BorderBlockFilename setting for the current drawing Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderBlockFilename is: " & dbPref.BorderBl vbInformation, "BorderBlockFilename Example"

BorderBottomMargin Example

Sub Example_BorderBottomMargin ()

' This example returns the BorderBottomMargin setting for the current drawing Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderBottomMargin is: " & dbPref.BorderBo vbInformation, "BorderBottomMargin Example"

BorderLeftMargin Example

Sub Example_BorderLeftMargin()

' This example returns the BorderLeftMargin setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderLeftMargin is: " & dbPref.BorderLeftM vbInformation, "BorderLeftMargin Example"

BorderLineWidth Example

Sub Example_BorderLineWidth()

' This example returns the BorderLineWidth setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderLineWidth is: " & dbPref.BorderLineW

BorderPath Example

Sub Example_BorderPath()

' This example returns the BorderPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for BorderPath is: " & prefFiles.BorderPath, _ vbInformation, "BorderPath Example"

BorderRightMargin Example

Sub Example_BorderRightMargin ()

' This example returns the BorderRightMargin setting for the current drawing. Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderRightMargin is: " & dbPref.BorderRigh

BorderStyle Example

Sub Example_BorderStyle()

```
'This example returns the BorderStyle setting for the current drawing.
Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.Preferences
'Convert the constant to a string.
Dim strStyle As String
If dbPref.BorderStyle = kBorderStyleLine Then
  strStyle = "line."
End If
If dbPref.BorderStyle = kBorderStyleUnscaledBlock Then
  strStyle = "unscaled block."
End If
If dbPref.BorderStyle = kBorderStyleScaledBlock Then
  strStyle = "scaled block."
End If
If dbPref.BorderStyle = kBorderStyleNone Then
  strStyle = "none."
End If
```

MsgBox "The current value for BorderStyle is " & strStyle, vbInformation, "Both

BorderTopMargin Example

Sub Example_BorderTopMargin ()

' This example returns the BorderTopMargin setting for the current drawing. Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for BorderTopMargin is: " & dbPref.BorderTopMargin vbInformation, "BorderTopMargin Example"

Boundaries Example

Sub Example_Boudaries()

'This example returns the number of Boundaries in the first

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of Boundaries in the first Surface is: " & surfIn.Boundaries, vbInformation, "Boundaries Example"

^{&#}x27; surface in the collection.

Boundary Example

Sub Example_Boundary()

'This example returns the first point in the Boundary

' for the first WaterShed in the collection.

Dim surf As AeccSurface

Dim wShed As AeccWaterShed

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wShed = surf.Outputs.WaterSheds.Item(0)

Dim coords As Variant

coords = wShed.Boundary

MsgBox "The first Boundary point for the first WaterShed is: " & coords(0) & , vbInformation, "Boundary Example"

Breaklines Example

Sub Example_SurfaceInputs_Breaklines()

'This example returns the number of Breaklines in the first

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of Breaklines in the first Surface is: " & surfIn.Breaklines, vbInformation, "Breaklines Example"

^{&#}x27; surface in the collection.

Build Example

Examples:

1 Surface

1 WaterSheds

```
Sub Example_Build_Surfaces()
```

```
'This example Builds the first Surface in the collection after adding 20.0
```

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

surf.AddToAllElevations 20# surf.Build

End Sub

Sub Example_Build_Watersheds()

 $\ '\ This\ example\ Builds\ the\ Watersheds\ after\ increasing\ the\ MinDepressionDeptl$

' by 10.0

Dim surf As AeccSurface

Dim wSheds As AeccWaterSheds

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wSheds = surf.Outputs.WaterSheds

Dim minDepth As Double
Dim minArea As Double

minDepth = wSheds.MinDepressionDepth + 10 minArea = wSheds.MinDepressionArea

wSheds.Build MinDepressionDepth, MinDepressionArea, True

^{&#}x27; to all elevations.

CCWFlag Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_CCWFlag_AlignCurve()
```

```
'This example returns the CCWFlag for the first Curve found in the
```

```
' first Alignment in the collection.
```

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

```
Dim alignMsg As String
```

alignMsg = "There is no Curve entity in the first Alignment."

```
' Find first Curve in the alignment
```

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then

alignMsg = "The CCWFlag for the first Curve in the alignment is: " & alignment is: "

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "CCWFlag Example"

End Sub

Sub Example_CCWflag_ParcelCurve()

- 'This example returns the CCWFlag for the first curve in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

```
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The CCWFlag for the first Curve in the Parcel is: " & parce
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "CCWFlag Example"
```

CenterEasting Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_CenterEasting_AlignCurve()
```

```
'This example returns the CenterEasting for the first Curve found in the
'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The CenterEasting for the first Curve in the alignment is: "
& Format(alignEnt.CenterEasting, "0.00")

Exit For

End If

Next
```

MsgBox alignMsg, vbInformation, "CenterEasting Example"

End Sub

Sub Example_CenterEasting_ParcelCurve()

- 'This example returns the CenterEasting for the first curve in the
- ' first Parcel in the collection.

```
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The CenterEasting for the first Curve in the Parcel is: " _
        & Format(parcelEnt.CenterEasting, "0.00")
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "CenterEasting Example"
```

CenterNorthing Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_CenterNorthing()
```

```
'This example returns the CenterNorthing for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The CenterNorthing for the first Curve in the alignment is: "
& Format(alignEnt.CenterNorthing, "0.00")
Exit For
End If
Next
```

MsgBox alignMsg, vbInformation, "CenterNorthing Example"

End Sub

Sub Example_CenterNorthing_ParcelCurve()

- 'This example returns the CenterNorthing for the first curve in the
- ' first Parcel in the collection.

```
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The CenterNorthing for the first Curve in the Parcel is: " _
        & Format(parcelEnt.CenterNorthing, "0.00")
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "CenterNorthing Example"
```

CentroidEasting Example

Sub Example_CentroidEasting()

```
'This example returns the CentroidEasting and the CentroidNorthing
```

Dim parcel As AeccParcel

Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

```
MsgBox "The CentroidEasting for the first Parcel in the collection is: " _ & Format(parcel.CentroidEasting, "0.00") & vbCrLf & _ "The CentroidNorthing for the first Parcel in the collection is: " _ & Format(parcel.CentroidNorthing, "0.00"), vbInformation, "CentroidEastin
```

^{&#}x27; for the first parcel in the collection.

CentroidNorthing Example

Sub Example_CentroidNorthing()

```
'This example returns the CentroidEasting and the CentroidNorthing
```

Dim parcel As AeccParcel

Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

```
MsgBox "The CentroidEasting for the first Parcel in the collection is: " _ & Format(parcel.CentroidEasting, "0.00") & vbCrLf & _ "The CentroidNorthing for the first Parcel in the collection is: " _ & Format(parcel.CentroidNorthing, "0.00"), vbInformation, "CentroidNorth
```

^{&#}x27; for the first parcel in the collection.

ChordDirection Example

Sub Example_ChordDirection()

```
'This example returns the ChordDirection for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The ChordDirection for the first Curve in the alignment is: "Exit For
End If

Next

MsgBox alignMsg, vbInformation, "ChordDirection Example"
```

ChordLength Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_ChordLength_AlignCurve()
```

```
'This example returns the ChordLength for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kCurve Then
        alignMsg = "The ChordLength for the first Curve in the alignment is: " _
        & Format(alignEnt.ChordLength, "0.00")
        Exit For
    End If
Next
```

MsgBox alignMsg, vbInformation, "ChordLength Example"

End Sub

Sub Example_ChordLength_ParcelCurve()

- 'This example returns the ChordLength for the first curve in the
- ' first Parcel in the collection.

```
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The ChordLength for the first Curve in the Parcel is: " _
        & Format(parcelEnt.ChordLength, "0.00")
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "ChordLength Example"
```

CivilDataFilesPath Example

Sub Example_CivilDataFilesPath()

' This example returns the CivilDataFilesPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for CivilDataFilesPath is: " & prefFiles.CivilDataI

ClearOverrides Example

Sub Example_ClearOverrides()

'This function clears the overrides for the first PoinGroup in

' the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

pntGrp.ClearOverrides

MsgBox "The overrides have been cleared for PointGroup: " & pntGrp.Group! vbInformation, "ClearOverrides Example"

Code Example

Examples:

1 <u>CrossSectionPointCode</u> (Civil Engineering Feature)

l DescriptionKey

Sub Example_Code_CrossSectionPointCode()

```
'This example returns the point code data for the first cross
```

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Dim xSectPCode As AeccCrossSectionPointCode

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

Set xSectPCode = xSect.CrossSectionPointCodes.Item(0)

' Get the alignment name

Dim alignName As String alignName = align.Name

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

```
MsgBox "The alignment name is: " & alignName & vbCrLf & _
"The first cross section is at station: " & station & vbCrLf & _
"The data for the first point code is: " & vbCrLf & _
vbTab & "Code: " & xSectPCode.Code & vbCrLf & _
vbTab & "Description: " & xSectPCode.Description & vbCrLf & _
vbTab & "Elevation: " & Format(xSectPCode.elevation, "0.00") & vbCrLf &
```

^{&#}x27; section surface in the collection for the first cross section

^{&#}x27; in the collection.

vbTab & "Offset: " & Format(xSectPCode.offset, "0.00"), _ vbInformation, "Code Example"

End Sub

Sub Example_Code_DescriptionKey()

- 'This example returns the Code for the first DescriptionKey
- ' in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey Code is " & dKey.Code, vbInformatio

Cogo Example

Sub Example_Cogo()

'This example returns the text size for Cogo Preferences

' in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The text size for Cogo Preferences in the current Project is: " & prefl , vbInformation, "Cogo Example"

CogoPoints Example

Sub Example_CogoPoints()

' This example returns the number of CogoPoints in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of CogoPoints in the current Project is: " & proj.CogoPovbInformation, "CogoPoints Example"

Composite Example

Sub Example_Composite()

'This example creates a composite surface from two existing surfaces.

Dim surfs As AeccSurfaces

Dim childSurf As AeccSurface

Dim parentSurf As AeccSurface

Dim newSurf As AeccSurface

Set surfs = AeccApplication.ActiveProject.Surfaces

Set childSurf = surfs.Item(0)

Set parentSurf = surfs.Item(1)

Set newSurf = surfs.Composite("New Surface", parentSurf.Name, childSurf.Name, childSurf.Name,

MsgBox "The number of faces in the new surface is: " & newSurf.NumberOfF vbInformation, "Composite Example"

ContourItems Example

Sub Example_ContourItems()

- 'This example returns the number of ContourItems in the first
- ' surface in the collection.

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of ContourItems in the first Surface is: " & surfIn.ContourItems Example"

ContourStyle Example

Sub Example_ContourStyle()

'This example displays the contour style for the selected contour.

```
On Error Resume Next
```

```
' Delete existing SelectionSet
ThisDrawing.SelectionSets("SSet").Delete
' Create the selection set based on a point selection
' and filter for Contour objects
Dim ssetObj As AcadSelectionSet
Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
Dim gpCode(0) As Integer
Dim dataValue(0) As Variant
gpCode(0) = 0
dataValue(0) = "AECC CONTOUR"
Dim groupCode As Variant
Dim dataCode As Variant
Dim returnPnt As Variant
groupCode = gpCode
dataCode = dataValue
returnPnt = ThisDrawing.Utility.GetPoint(, "Select a contour line: ")
```

ssetObj.SelectAtPoint returnPnt, groupCode, dataCode

Dim objContour As AeccContour Set objContour = ssetObj.Item(0)

MsgBox "The ContourStyle for the selected contour is: " & objContour.Contour vbInformation, "ContourStyle Example"

ContourStyleName Example

Sub Example_ContourStyleName()

'This example displays the ContourStyleName for a selected contour

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

'Create the selection set based on a point selection

' and filter for Contour objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim mode As Integer Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AECC CONTOUR"

Dim groupCode As Variant Dim dataCode As Variant Dim returnPnt As Variant

groupCode = gpCode dataCode = dataValue

returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point on a contour line: ")
ssetObj.SelectAtPoint returnPnt, groupCode, dataCode

Dim objContour As AeccContour
Set objContour = ssetObj.Item(0)

MsgBox "The ContourStyleName for the selected contour is " & objContour.C vbInformation, "ContourStyleName Example"

ContourStylesPath Example

Sub Example_ContourStylesPath()

' This example returns the ContourStylesPath setting. Dim prefFiles As AeccPreferencesFiles Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for ContourStylesPath is: " & prefFiles.ContourSty vbInformation, "ContourStylesPath Example"

ConvertToCurrentAreaDisplay Example

Sub Example_ConvertToCurrentAreaDisplay()

```
'This example returns the value of ConvertToCurrentAreaDisplay
```

' for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

' Set the area display units to square yards dbPref.AreaDisplayUnit = aecUnitSquareYard

Dim SourceArea As Double Dim TargetArea As Double

SourceArea = 36# 'Square feet, assumes drawing set to foot units TargetArea = dbPref.ConvertToCurrentAreaDisplay(SourceArea)

MsgBox "The area to convert is 36 square feet." & vbCrLf & _ "The AreaDisplayUnit is set to square yard." & vbCrLf & _ "The value for ConvertToCurrentAreaDisplay is " & TargetArea & " square vbInformation, "ConvertToCurrentAreaDisplay Example"

ConvertToCurrentVolumeDisplay Example

Sub Example_ConvertToCurrentVolumeDisplay()

```
'This example returns the value of ConvertToCurrentVolumeDisplay
```

Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences

' Set the volume display units to cubic yards dbPref.VolumeDisplayUnit = aecUnitCubicYard

Dim SourceArea As Double Dim TargetArea As Double

SourceArea = 54# 'Cubic feet, assumes drawing set to foot units TargetArea = dbPref.ConvertToCurrentVolumeDisplay(SourceArea)

MsgBox "The area to convert is 54 cubic feet." & vbCrLf & _ "The VolumnDisplayUnit is set to cubic yard." & vbCrLf & _ "The value for ConvertToCurrentVolumeDisplay is " & TargetArea & " squa vbInformation, "ConvertToCurrentVolumnDisplay Example"

^{&#}x27; for the current drawing.

CoordinatePrecision Example

Sub Example_CoordinatePrecision()

' This example returns the CoordinatePrecision setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for CoordinatePrecision is: " & dbPref. Coordinate vbInformation, "CoordinatePrecision Example"

Coordinates Example

Examples:

- 1 AeccContour
- 1 **Boundary**
- 1 Breakline
- l CogoPoint
- 1 ContourItem
- 1 CrossSectionBlock (Civil Engineering Feature)
- 1 Edge
- 1 ElevationContour
- l Face
- 1 ProfileBlock (Civil Engineering Feature)

Sub Example_Coordinates_AeccContour()

'This example displays the Coordinates for a selected contour

On Error Resume Next

- ' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete
- ' Create the selection set based on a point selection
- ' and filter for Contour objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

```
Dim mode As Integer
Dim gpCode(0) As Integer
Dim dataValue(0) As Variant
gpCode(0) = 0
dataValue(0) = "AECC_CONTOUR"
Dim groupCode As Variant
Dim dataCode As Variant
Dim returnPnt As Variant
groupCode = gpCode
dataCode = dataValue
returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point on a contour line: ")
ssetObj.SelectAtPoint returnPnt, groupCode, dataCode
Dim objContour As AeccContour
Set objContour = ssetObj.Item(0)
Dim coords As Variant
coords = objContour.Coordinates
```

MsgBox "The first point in Coordinates is: " & coords(0) & ", " & coords(1) & vbInformation, "Coordinates Example"

End Sub

Sub Example_Coordinates_Boundary()

Dim surf As AeccSurface

Dim bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bound = surf.Inputs.Boundaries.Item(0)

Dim coords As Variant

^{&#}x27;This example returns the first point in the Coordinates

^{&#}x27; for the first Boundary in the collection.

coords = bound.Coordinates

MsgBox "The first Coordinate for the Boundary is: " & coords(0) & ", " & coo , vbInformation, "Coordinates Example"

End Sub

Sub Example_Coordinates_BreakLine()

- 'This example returns the first point in the Coordinates
- ' for the first BreakLine in the collection.

Dim surf As AeccSurface

Dim brkLine As AeccBreakLine

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set brkLine = surf.Inputs.BreakLines.Item(0)

Dim coords As Variant

coords = brkLine.Coordinates

MsgBox "The first Coordinate for the BreakLine is: " & coords(0) & ", " & coords, vbInformation, "Coordinates Example"

End Sub

Sub Example_Coordinates_CogoPoint()

'This example returns the Coordinate for the first CogoPoint in the collection Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

Dim coords As Variant

coords = cogoPnt.Coordinates

MsgBox "The Coordinates for the first CogoPoint in the collection is: " & coor

End Sub

Sub Example_Coordinates_ContourItem()

'This example returns the first point in the Coordinates

' for the first ContourItem in the collection.

Dim surf As AeccSurface

Dim cont As AeccContourItem

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set cont = Surf.Inputs.ContourItems.Item(0)

Dim coords As Variant

coords = cont.Coordinates

MsgBox "The first Coordinates for the ContourItem is: " & coords(0) & ", " & , vbInformation, "Coordinates Example"

End Sub

Sub Example_Coordinates_CrossSectionBlock()

- 'This example returns the coordinates for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

'Get the coordinates for the first alignment cross section in the collection Dim coords As Variant coords = alignXSect.Coordinates

```
MsgBox "The coordinates for the alignment cross section at station " & station "X Value: " & Format(coords(0), "0.00") & vbCrLf & _ "Y Value: " & Format(coords(1), "0.00"), vbInformation, "Coordinates Exan
```

End Sub

```
Sub Example_Coordinates_Edge()
```

```
'This example returns the coordinates for the first
'Edge in the collection.

Dim surf As AeccSurface
Dim edge As AeccEdge
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set edge = Surf.Outputs.Edges.Item(0)

Dim coords As Variant
coords = edge.Coordinates

MsgBox "The Coordinates for the first Edge are:" & vbCrLf & _
    "FromNorthing: " & coords(0) & vbCrLf & _
    "FromEasting: " & coords(1) & vbCrLf & _
    "FromElevation: " & coords(2) & vbCrLf & _
    "ToNorthing: " & coords(3) & vbCrLf & _
    "ToNorthing: " & coords(3) & vbCrLf & _
```

End Sub

Sub Example_Coordinates_ElevationContour()

"ToEasting: " & coords(4) & vbCrLf & _

'This example returns the first point in the coordinates for the

"ToElevation: " & coords(5), vbInformation, "Coordinates Example"

' first ElevationContour in the collection at 100.0

Dim surf As AeccSurface

Dim elevContours As AeccElevationContours

Dim elevContour As AeccElevationContour

Dim coords As Variant

```
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Set elevContours = Surf.Outputs.ElevationContours
  'Create a collection of ElevationContours at 100.0
  elevContours.Elevation = 100#
  Set elevContour = elevContours.Item(0)
  coords = elevContour.Coordinates
  MsgBox "The first point in Coordinates is: " & coords(0) & ", " & coords(1) &
End Sub
Sub Example_Coordinates_Face()
  'This example returns the Coordinates
  ' for the first Face in the collection.
  Dim surf As AeccSurface
  Dim face As AeccFace
  Set Surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Set face = surf.Outputs.faces.Item(0)
  Dim coord As Variant
  coord = face.Coordinates
  MsgBox "The points for the first face are: " & vbCrLf & _
    coord(0) & ", " & coord(1) & ", " & coord(2) & vbCrLf & _
    coord(3) & ", " & coord(4) & ", " & coord(5) & vbCrLf & _
    coord(6) & ", " & coord(7) & ", " & coord(8), vbInformation, "Coordinates ]
End Sub
```

Sub Example_Coordinates_ProfileBlock()

- 'This example returns the Coordinate for the first ProfileBlock
- ' in the collection.

Dim alignProf As AeccProfileBlock
Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

Dim coords As Variant coords = alignProf.Coordinates

MsgBox "The Coordinates for the first Alignment Profile are:" & vbCrLf & _ "X Value: " & Format(coords(0), "0.00") & vbCrLf & _ "Y Value: " & Format(coords(1), "0.00"), vbInformation, "Coordinates Exan

CoordinateZone Example

Examples:

1 DatabasePreferences

1 DEMFile

Sub Example_CoordinateZone_DatabasePreferences()

'This example returns the CoordinateZone setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for CoordinateZone is: " & dbPref.CoordinateZone vbInformation, "CoordinateZone Example"

End Sub

Sub Example_CoordinateZone_DEMFile()

- 'This example returns the CoordinateZone for the first DEMFile in the
- ' first surface in the collection.

Dim surf As AeccSurface

Dim DEMFile As AeccDEMFile

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set DEMFile = surf.Inputs.DEMFiles.Item(0)

MsgBox "The CoordinateZone for the first DEMFile is: " & DEMFile.Coordin vbInformation, "CoordinateZone Example"

Copy Example

Examples

1 Prototypes

l Surfaces

```
Sub Example_Copy_Prototypes()
```

```
'This example copies a the first prototype in the collection
```

Dim prots As AeccPrototypes

Dim prot As AeccPrototype

Dim protcopy As AeccPrototype

Set prots = AeccApplication.Prototypes

Set prot = prots.Item(0)

Dim source As String
Dim target As String

'Get the source and target prototype names

source = prot.Name
target = "Copy of " & source

'Make copy of prototype

Set protcopy = prots.Copy(source, target)

MsgBox "The copied prototype is named: " & protcopy.Name, vbInformation,

End Sub

Sub Example_Copy_Surfaces()

'This example copies a surface.

Dim surfs As AeccSurfaces

^{&#}x27; to a "Copy of " the first prototype..

Dim surf As AeccSurface Dim surfCopy As AeccSurface Set surfs = AeccApplication.ActiveProject.Surfaces Set surf = surfs.Item(0)

Dim source As String

' Get name of surface to copy source = surf.Name

Set surfCopy = surfs.Copy(source)

MsgBox "The copied surface is named: " & surfCopy.Name, vbInformation, "(

Count Example

Sub Example_Count()

- ' Use count to retrieve the number of entities in a collection.
- 'You might use this value in a loop structure to iterate through the collection. MsgBox "There are " & AeccApplication.Projects.count & " project(s) at the p vbInformation, "Count Example"

```
Dim objCount As Integer
objCount = AeccApplication.Projects.count

Dim I As Integer
Dim proj As AeccProject
For I = 0 To objCount - 1
    Set proj = AeccApplication.Projects.Item(I)
    MsgBox "The projects collection includes: " & proj.Name, _
    vbInformation, "Count Example"
Next
```

Course Example

Sub Example_Course()

- 'This example returns the Course for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The Course of the first entity in the Parcel is: " & parcelEnt.Course, vbInformation, "Course Example"

CourseIn Example

```
Sub Example_CourseIn()
```

```
'This example returns the CourseIn for the first curve in the
' first Parcel in the collection.
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)
Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."
' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
  If parcelEnt.Type = kParcelCurve Then
    parcelMsg = "The CourseIn for the first Curve in the Parcel is: "_
    & parcelEnt.CourseIn & vbCrLf _
    & "The CourseOut for the first Curve in the Parcel is: " _
    & parcelEnt.CourseOut
  Exit For
  End If
Next
MsgBox parcelMsg, vbInformation, "CourseIn Example"
```

CourseOut Example

```
Sub Example_CourseOut()
```

```
'This example returns the CourseOut for the first curve in the
' first Parcel in the collection.
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)
Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."
' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
  If parcelEnt.Type = kParcelCurve Then
    parcelMsg = "The CourseIn for the first Curve in the Parcel is: "_
    & parcelEnt.CourseIn & vbCrLf _
    & "The CourseOut for the first Curve in the Parcel is: " _
    & parcelEnt.CourseOut
  Exit For
  End If
Next
MsgBox parcelMsg, vbInformation, "CourseOut Example"
```

CrossSection Example

Sub Example_CrossSection()

- 'This example returns the datum layer for the cross section
- ' preferences in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The datum layer for cross section preferences in the current project in the prefPrj.CrossSection.GetString(kDatumLayer), vbInformation, "Profile Edition of the Company of the Com

CrossSectionBlocks Example

Sub Example_CrossSectionBlocks()

- 'This example returns the count of cross section blocks in
- ' the current document.

Dim doc As AeccDocument

Set doc = AeccApplication.ActiveDocument

MsgBox "The number of cross section blocks in the current document is " & _ doc.CrossSectionBlocks.Count, vbInformation, "CrossSectionBlocks Examp

CrossSectionPointCodes Example

Sub Example_CrossSectionPointCodes()

- 'This example returns the count of point codes for the
- ' first cross section in the collection.

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set xSect = align.CrossSections.Item(0)

MsgBox "The number of point codes in the first cross section is: "_ & xSect.CrossSectionPointCodes.Count, vbInformation, "CrossSectionPointCodes.Count, vbInform

CrossSections Example

Sub Example_CrossSections()

- 'This example returns the number of cross sections for the
- ' first alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The number of cross sections in first alignment is: " _ & align.CrossSections.Count , vbInformation, "CrossSections Example"

CrossSectionSurfaces Example

Sub Example_CrossSectionSurfaces()

- 'This example returns the count of surfaces for the
- ' first cross section in the collection.

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set xSect = align.CrossSections.Item(0)

MsgBox "The number of surfaces in the first cross section is: "_ & xSect.CrossSectionSurfaces.Count, vbInformation, "CrossSectionSurfaces.

CurrentAlignment Example

Sub Example_CurrentAlignment()

'This example returns the CurrentAlignment Value.

Dim aligns As AeccAlignments

Set aligns = AeccApplication.ActiveProject.Alignments

MsgBox "The CurrentAlignment for alignments is " & aligns.CurrentAlignment vbInformation, "CurrentAlignment Example"

CurrentSurface Example

Sub Example_CurrentSurface()

'This example returns the CurrentSurface setting.

Dim surfs As AeccSurfaces

Set surfs = AeccApplication.ActiveProject.Surfaces

MsgBox "The CurrentSurface is: " & surfs.CurrentSurface, vbInformation, "Cı

CurveCode Example

Sub Example_CurveCode()

- 'This example returns the data for the first superelevation in the
- ' superelevation collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim sElev As AeccSuperelevation

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set sElev = align.Superelevations.Item(0)

' Get the alignment name Dim alignName As String alignName = align.Name

' Get the superelevation station and format it Dim station As String station = aligns.DoubleToStaFormat(sElev.station)

' Get the superelevation curve code

Dim crvCode As String

Select Case sElev.curveCode

Case kSERightHandCurve

crvCode = "Right Hand Curve"

Case kSELeftHandCurve

crvCode = "Left Hand Curve"

End Select

' Get the superelevation code

Dim supCode As String

Select Case sElev.SuperelevationCode

Case kSEFullCrown

supCode = "Full Crown"

Case kSEHalfCrown

```
supCode = "Half Crown"
Case kSECrownRemoved
supCode = "Crown Removed"
Case kSEFullSuperelevations
supCode = "Full Superelevation"
Case kSEReverseCurve
supCode = "Reverse Curve"
Case kSECompoundCurve
supCode = "Compound Curve"
End Select
```

```
MsgBox "The alignment name is: " & alignName & vbCrLf & _
"The data for the first superelevation is: " & vbCrLf & _
vbTab & "Station: " & station & vbCrLf & _
vbTab & "Curve Code: " & crvCode & vbCrLf & _
vbTab & "Superelevation Code: " & supCode, _
vbInformation, "CurveCode Example"
```

CurveLabelIncrement Example

Sub Example_CurveLabelIncrement()

'This example returns the CurveLabelIncrement for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The CurveLabelIncrement for the first ProfileBlock in the collection & alignProf.CurveLabelIncrement, vbInformation, "CurveLabelIncrement I

CurveLength Example

Sub Example_CurveLength()

```
'This example returns the CurveLength value for the second PVI in the
```

' first finished ground of the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Dim PVI As AeccPVI

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

Set PVI = FGProf.PVIs.Item(1)

MsgBox "The CurveLength of the second PVI is: "_ & Format(PVI.CurveLength, "0.000"), vbInformation, "CurveLength Examı

DatabaseScale Example

Sub Example_DatabaseScale()

'This example returns the HorizontalScale setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for DatabaseScale is: " & dbPref.DatabaseScale, _ vbInformation, "DatabaseScale Example"

DataPath Example

Sub Example_DataPath()

' This example returns the DataPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for DataPath is: " & prefFiles.DataPath, _ vbInformation, "DataPath Example"

DatumElevation Example

Examples:

- 1 <u>CrossSectionBlock</u> (Civil Engineering Feature)
- 1 ProfileBlock (Civil Engineering Feature)

```
Sub
Example_DatumElevation_CrossSectionBlock()
This example returns the datum elevation for the
first alignment cross section in the collection.
  Dim
alignXSects As AeccCrossSectionBlocks
  Dim
alignXSect As AeccCrossSectionBlock
alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks
  Set
alignXSect = alignXSects.Item(0)
the station for the first alignment cross section in the collection
  Dim
station As String
  station
= alignXSect.station
  MsgBox
"The datum elevation for the alignment cross section at station " & station &
" is: " &
    Format(alignXSect.DatumElevation,
"0.00"), vbInformation, "DatumElevation Example"
```

Sub

```
Sub
Example_DatumElevation_ProfileBlock()

'
This example returns the DatumElevation for the first ProfileBlock
in the collection
Dim
alignProf As AeccProfileBlock
Set
alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox
"The DatumElevation for the first ProfileBlock in the collection is: "__
& alignProf.DatumElevation, vbInformation, "DatumElevation Example"
End
```

Delete Example

Examples:

- l Alignments
- 1 **Boundaries**
- 1 BreakLines
- l CogoPoints
- l ContourItems
- 1 **DEMFiles**
- 1 <u>DescriptionKeyFile</u>
- 1 <u>DescriptionKeyFiles</u>
- l <u>EGProfiles</u> (Civil Engineering Feature)
- 1 <u>FGProfiles</u> (Civil Engineering Feature)
- 1 Parcels
- l <u>PointFiles</u>
- l <u>PointGroupNames</u>
- l <u>PointGroups</u>
- l Projects
- l Prototypes
- l <u>PVIs</u> (Civil Engineering Feature)
- l StationEquations

1 Surfaces

Sub Example_Delete_Alignments()

'This example deletes an Alignment from the Alignments Collection

Dim aligns As AeccAlignments

Set aligns = AeccApplication.ActiveProject.Alignments

' Show the number of alignments in the project

MsgBox "The initial Count of Alignments is: " & aligns.Count, vbInformation,

- ' Add an Alignment named New Alignment and starting at Station 2000 aligns. Add "New Alignment", 2000
- ' Show the number of alignments in the project
 MsgBox "The Count of Alignments after Add is: " & aligns.Count, vbInformat
- ' Deletes the Alignment named New Alignment aligns. Delete "New Alignment"
- 'Show the number of alignments in the project
 MsgBox "The Count of Alignments after Delete is: " & aligns.Count, vbInform

End Sub

Sub Example_Delete_Boundaries()

- 'This example starts by displays the initial count of the Boundaries
- ' for the first surface in the collection. A new visible Boundary is added
- ' and the count is redisplays. Finally, the new Boundary is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim bounds As AeccBoundaries

Dim Bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bounds = surf.Inputs.Boundaries

MsgBox "The Boundaries Count is: " & bounds.count, vbInformation, "Delete

```
' Initialize variables
Dim pnts(0 To 14) As Double
Dim count As Integer
Dim index As Integer
Dim pnt As Variant
index = 0
```

' Add a new Boundary based on five selected points

```
For count = 1 To 5

pnt = ThisDrawing.Utility.GetPoint(, "Select point" + Str(count) + " of Bour pnts(index) = pnt(0): pnts(index + 1) = pnt(1): pnts(index + 2) = pnt(2) index = index + 3
```

Next count

' Add a new Boundary

Set Bound = bounds.Add(kBoundaryTypeShow, False, pnts, "NewBoundary")

MsgBox "The BreakLines Count after the add is: " & bounds.count, vbInforma

' Delete the new Boundary bounds.Delete Bound.Id

MsgBox "The Boundaries Count after the delete is " & bounds.count, vbInforn

End Sub

Sub Example_Delete_BreakLines()

- 'This example starts by displays the initial count of the BreakLines
- ' for the first surface in the collection. A new BreakLines is added
- ' and the count is redisplays. Finally, the new file is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim brkLines As AeccBreakLines

```
Dim brkLine As AeccBreakLine
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set brkLines = surf.Inputs.BreakLines
```

MsgBox "The BreakLines Count is: " & brkLines.count, vbInformation, "Dele

```
' Initialize variables
Dim pnts(0 To 14) As Double
Dim count As Integer
Dim index As Integer
Dim pnt As Variant
```

index = 0

' Add a new BreakLine based on five selected points

```
For count = 1 To 5

pnt = ThisDrawing.Utility.GetPoint(, "Select point" + Str(count) + " of Breal pnts(index) = pnt(0): pnts(index + 1) = pnt(1): pnts(index + 2) = pnt(2) index = index + 3
```

Next count

Set brkLine = brkLines.Add(pnts, "NewBreakLine")

MsgBox "The BreakLines Count after the add is: " & brkLines.count, vbInforn

' Delete the new BreakLines

brkLines.Delete brkLine.Id

MsgBox "The BreakLines Count after the delete is " & brkLines.count, vbInfo

End Sub

Sub Example_Delete_CogoPoints()

Dim cogoPnts As AeccCogoPoints

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

^{&#}x27;This examples displays the used point numbers, deletes a point number

^{&#}x27; and the displays the used point numbers again to confirm the deletion.

```
MsgBox "Used point numbers are " & cogoPnts.UsedPointNumbers, _ vbInformation, "Delete Example"
```

' Get a point number

Dim pntNum As Long

pntNum = ThisDrawing.Utility.GetReal("Enter a valid point number : ")
pntNum = Fix(pntNum)

' Delete the CogoPoint cogoPnts.Delete pntNum

MsgBox "Used point numbers are " & cogoPnts.UsedPointNumbers, _ vbInformation, "Delete Example"

End Sub

Sub Example_Delete_ContourItems()

- 'This example Deletes the ContourItem with ID equal to one (1)
- ' from the first collection of ContourItems.

Dim surf As AeccSurface

Dim conts As AeccContourItems

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set conts = surf.Inputs.ContourItems

- 'Display the number of ContourItems in the collection before the delete MsgBox "The number of ContourItems in the collection is: " & conts.Count, _ vbInformation, "Delete Example"
- ' Delete ContourItem with ID equal to one. conts.Delete 1
- 'Display the number of ContourItems in the collection after the delete MsgBox "The number of ContourItems in the collection is: " & conts.Count, _ vbInformation, "Delete Example"

End Sub

Sub Example_Delete_DEMFiles()

- 'This example starts by displays the initial count of DEMFiles
- ' for the first surface. A new DEMfile is added and the count is redisplays.
- ' Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim surf As AeccSurface

Dim DEMFiles As AeccDEMFiles

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set DEMFiles = surf.Inputs.DEMFiles

MsgBox "The DEMFiles Count is: " & DEMFiles.Count, vbInformation, "Delo

'Get the DEMFile name and format the prompt

Dim DEMName As String

Dim prompt As String

prompt = "Enter the name of the DEMFile for surface " & surf.Name & ": "
DEMName = ThisDrawing.Utility.GetString(False, prompt)

' Add a new DEMFile DEMFiles.Add DEMName

MsgBox "The DEMFiles Count after the add is: " & DEMFiles.Count, vbInfor

' Delete the new DEMFile DEMFiles.Delete DEMName

MsgBox "The DEMFiles Count after the delete is " & DEMFiles.Count, vbInfo

End Sub

Sub Example_Delete_DescriptionKeyFile()

'This example starts by displays the initial count of DescriptionKeys

Dim dkeyFile As AeccDescriptionKeyFile

Set dkeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item("DEI

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "D

' Add a new DescriptionKey dkeyFile.Add "New"

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "D

' Delete the new DescriptionKey dkeyFile.Delete "New"

MsgBox "The DescriptionKey Count is " & dkeyFile.count, vbInformation, "D

End Sub

Sub Example_Delete_DescriptionKeyFiles()

- 'This example starts by displays the initial count of DescriptionKeyFiles
- ' on the system. A new DescriptionKeyFile is added and the count is redisplays
- ' Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim dKeyFiles As AeccDescriptionKeyFiles
Set dKeyFiles = AeccApplication.ActiveProject.DescriptionKeyFiles

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

' Add a new DescriptionKeyFile dKeyFiles.Add "NewFile"

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

' Delete the new DescriptionKeyFile

^{&#}x27; in the DEFAULT DescriptionKey file. A new DescriptionKey is added

^{&#}x27; and the count is redisplays. Finally, the new key is deleted. The

^{&#}x27; count is displayed again, showing the deletion.

dKeyFiles.Delete "NewFile"

MsgBox "The DescriptionKeyFiles Count is " & dKeyFiles.count, vbInformati

End Sub

Sub Example_Delete_EGProfiles()

'This example deletes the first existing ground profile

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

MsgBox "The number of existing ground profiles in the first alignment is: " _ & align.EGProfiles.Count, vbInformation, "Delete Example"

Dim surfName As String Dim offset As Integer surfName = EGProf.SurfaceName offset = EGProf.Type

' Delete the profile align.EGProfiles.Delete offset, surfName

MsgBox "The number of existing ground profiles in the first alignment is: " _ & align.EGProfiles.Count, vbInformation, "Delete Example"

End Sub

Sub Example_Delete_FGProfiles()

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

^{&#}x27; in the first alignment in the collection.

^{&#}x27;This example deletes the first finished ground profile

^{&#}x27; in the first alignment in the collection.

Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Set FGProf = align.FGProfiles.Item(0)

MsgBox "The number of finished ground profiles in the first alignment is: " _ & align.fGProfiles.Count, vbInformation, "Delete Example"

Dim offset As Integer offset = FGProf.Type

' Delete the profile align.FGProfiles.Delete offset

MsgBox "The number of finished ground profiles in the first alignment is: " _ & align.FGProfiles.Count, vbInformation, "Delete Example"

End Sub

Sub Example_Delete_Parcels()

- 'This example starts by displaying the initial count of Parcels
- ' A new Parcel is added and the count is redisplayed. Finally,
- ' the new Parcel is deleted. The count is displayed again, showing
- ' the deletion.

Dim parcels As AeccParcels
Set parcels = AeccApplication.ActiveProject.Parcels

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

' Add a new Parcel parcels. Add "NewParcel"

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

' Delete the new Parcel parcels.Delete "NewParcel"

MsgBox "The Parcel Count is: " & parcels.Count, vbInformation, "Add Examp

End Sub

Sub Example_Delete_PointFiles()

- 'This example starts by displays the initial count of PointFiles
- ' on the system. A new Pointfile is added and the count is redisplays.
- 'Finally, the new file is deleted. The count is displayed again, showing
- ' the deletion.

Dim surf As AeccSurface

Dim pntFiles As AeccPointFiles

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set pntFiles = surf.Inputs.PointFiles

MsgBox "The PointFiles Count is: " & pntFiles.Count, vbInformation, "Delete

' Add a new PointFile pntFiles.Add "NewPointFile"

MsgBox "The PointFiles Count after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add is: " & pntFiles.Count, vbInformation and its point after the add its point after the add

' Delete the new PointFile pntFiles.Delete "NewPointFile"

MsgBox "The PointFiles Count after the delete is " & pntFiles.Count, vbInforn

End Sub

Sub Example_Delete_PointGroupNames()

- 'This example starts by displays the initial count of the PointGroupNames
- ' for the first surface in the collection. A new PointGroupName is added
- ' and the count is redisplays. Finally, the new file is deleted. The count is
- ' displayed again, showing the deletion.

Dim surf As AeccSurface

Dim pntGrpNames As AeccPointGroupNames

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set pntGrpNames = surf.Inputs.PointGroupNames

MsgBox "The PointGroupNames Count is: " & pntGrpNames.Count, vbInform

' Add a new PointGroupName pntGrpNames.Add "NewPointGroupName"

MsgBox "The PointGroupNames Count after the add is: " & pntGrpNames.Co

' Delete the new PointGroupName pntGrpNames.Delete "NewPointGroupName"

MsgBox "The PointGroupNames Count after the delete is " & pntGrpNames.C

End Sub

Sub Example_Delete_Projects()

- 'This example creates a new project named "New Project" and then
- ' deletes the project. A project count is displayed during the process.

Dim projs As AeccProjects

Set projs = AeccApplication.Projects

'Show the number of projects
MsgBox "The initial Count of Projects is: " & projs.Count, vbInformation, "De projs.Add "New Project", "Default (Feet)"

' Show the number of projects after the Add MsgBox "The Count of Projects after Add is: " & projs.Count, vbInformation, projs.Delete "New Project"

' Show the number of projects after the Delete
MsgBox "The Count of Projects after Delete is: " & projs.Count, vbInformation

Sub Example_Delete_Prototypes()

'This example deletes the first prototype in the collection.

Dim prots As AeccPrototypes

Dim prot As AeccPrototype

Set prots = AeccApplication.Prototypes

Set prot = prots.Item(0)

' Get the source prototype names

Dim source As String source = prot.Name

' Delete the prototype

prots.Delete source

MsgBox "The deleted prototype was named: " & source, vbInformation, "Dele

End Sub

Sub Example_Delete_PointGroups()

'This example adds and deletes a PointGroup from the PoinGroups Collection.

Dim pntGrps As AeccPointGroups

Set pntGrps = AeccApplication.ActiveProject.PointGroups

' Show the number of PointGroups in the project MsgBox "The initial Count of PointGroups is: " & pntGrps.Count, vbInformati

' Add PointGroup pntGrps.Add "New Group", "1-100"

' Show the number of PointGroups in the project after the Add MsgBox "The Count of PointGroups is: " & pntGrps.Count, vbInformation, "E

' Deletes the PointGroup named "New Group"

pntGrps.Delete "New Group"
' Show the number of PointGroups in the project after the Delete
MsgBox "The Count of PointGroups is: " & pntGrps.Count, vbInformation, "E

End Sub

Sub Example_Delete_PVIs()

'This example deletes the second PVI in the first finished

' ground profile of the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Dim PVI As AeccPVI

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

Set PVI = FGProf.PVIs.Item(1)

MsgBox "The number of PVIs in the finished ground profile is: " _ & FGProf.PVIs.Count, vbInformation, "Delete Example"

'Get the station and the elevation

Dim station As Double

station = PVI.station

' Delete the second PVI

FGProf.PVIs.Delete station

MsgBox "The number of PVIs in the finished ground profile is: " _ & FGProf.PVIs.Count, vbInformation, "Delete Example"

End Sub

Sub Example_Delete_StationEquations()

Dim aligns As AeccAlignments

^{&#}x27;This example deletes the first StationEquations from

^{&#}x27; the first alignment in the collection.

Dim align As AeccAlignment Set aligns = AeccApplication.ActiveProject.Alignments

' Get the first alignment Set align = aligns.Item(0)

' Set the first alignment current. aligns.CurrentAlignment = align.Name

Dim staEqus As AeccStationEquations Set staEqus = align.StationEquations

- 'Show the StationEquation count for the first alignment before the delete MsgBox "The StationEquation count for the first alignment is: " & staEqus.cou vbInformation, "Delete Example"
- ' Delete the first StationEquation staEqus.Delete(0)
- 'Show the StationEquation count for the first alignment after the delete MsgBox "The StationEquation count for the first alignment is: " & staEqus.couvbInformation, "Delete Example"

End Sub

Sub Example_Delete_Surfaces()

'This example deletes the first surface from the collection.

Dim surfs As AeccSurfaces

Dim surf As AeccSurface

Set surfs = AeccApplication.ActiveProject.Surfaces

Set surf = surfs.Item(0)

' Get the name of the first surface in the collection

Dim surfName As String surfName = surf.Name

- ' Show the Surfaces count before the delete MsgBox "The Surface count is: " & surfs.Count, vbInformation, "Delete Exam
- ' Delete the first StationEquation surfs.Delete (surfName)
- ' Show the Surfaces count after the delete MsgBox "The Surfaces count is: " & surfs.Count, vbInformation, "Delete Exar

Delta Example

Examples:

- 1 AlignCurve
- 1 AlignSpiral
- 1 ParcelCurve

```
Sub Example_Delta_AlignCurve()
```

```
'This example returns the Delta for the first Curve found in the
'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then

alignMsg = "The Delta for the first Curve in the alignment is: "_
& Format(alignEnt.Delta, "0.00")

Exit For

End If

Next
```

MsgBox alignMsg, vbInformation, "Delta Example"

End Sub

Sub Example_Delta_AlignSpiral()

'This example returns the Delta for the first Spiral found in the

```
' first Alignment in the collection.
  Dim align As AeccAlignment
  Dim alignEnts As AeccAlignEntity
  Set align = AeccApplication.ActiveProject.Alignments.Item(0)
  Dim alignMsg As String
  alignMsg = "There is no Spiral entity in the first Alignment."
  ' Find first Spiral in the alignment
  For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
       alignMsg = "The Delta for the first Spiral in the alignment is: "_
       & Format(alignEnt.Delta, "0.00")
       Exit For
    End If
  Next
  MsgBox alignMsg, vbInformation, "Delta Example"
End Sub
Sub Example_Delta_ParcelCurve()
  'This example returns the Delta for the first curve in the
  ' first Parcel in the collection.
  Dim parcel As AeccParcel
  Dim parcelEnt As AeccParcelEntity
  Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
  Set parcelEnt = parcel.ParcelEntities.Item(0)
  Dim parcelMsg As String
  parcelMsg = "There is no Curve entity in the first Parcel."
  ' Find first Curve in the parcel
  For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
       parcelMsg = "The Delta for the first Curve in the Parcel is: "
```

```
& Format(parcelEnt.Delta, "0.00")
Exit For
End If
Next
```

MsgBox parcelMsg, vbInformation, "Delta Example"

DEMFiles Example

Sub Example_DEMFiles()

'This example returns the number of DEMFiles in the first

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of DEMFiles in the first Surface is: " & surfIn.DEMFile , vbInformation, "DEMFiles Example"

^{&#}x27; surface in the collection.

Description Example

Examples:

- l AeccPoint
- 1 Alignment
- l <u>Boundary</u>
- 1 BreakLine
- 1 <u>CrossSectionPointCode</u> (Civil Engineering Feature)
- l PointGroup
- 1 Project
- l Prototype
- 1 Surface

Sub Example_Description_AeccPoint()

- 'This example returns the Description setting for the
- ' first Point object in a selection set

On Error Resume Next

- ' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete
- ' Create the selection set based on a point selection
- ' and filter for the Point objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer

```
Dim dataValue(0) As Variant
```

```
gpCode(0) = 0
dataValue(0) = "AECC_POINT"
```

Dim groupCode As Variant Dim dataCode As Variant

ThisDrawing.Utility.Prompt vbCrLf & "Select a point object..." groupCode = gpCode dataCode = dataValue ssetObj.SelectOnScreen groupCode, dataCode

Dim objPoint As AeccPoint Set objPoint = ssetObj.Item(0)

MsgBox "The value for Description is: " & objPoint.Description, _ vbInformation, "Description Example"

End Sub

Sub Example_Description_Alignment()

'This example returns the Description for the first Alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The Description of the first Alignment is: " & align.Description, _ vbInformation, "Description Example"

End Sub

Sub Example_Description_Boundary()

- 'This example returns the Description
- ' for the first Boundary in the collection.

Dim surf As AeccSurface

Dim bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bound = surf.Inputs.Boundaries.Item(0)

MsgBox "The Description for the Boundary is: " & bound.Description, _ vbInformation, "Description Example"

End Sub

Sub Example_Description_BreakLine()

- 'This example returns the Description
- ' for the first BreakLine in the collection.

Dim surf As AeccSurface

Dim brkLine As AeccBreakLine

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set brkLine = surf.Inputs.BreakLines.Item(0)

MsgBox "The Description for the BreakLine is: " & brkLine.Description, _ vbInformation, "Description Example"

End Sub

Sub Example_Description_CrossSectionPointCode()

- 'This example returns the point code data for the first cross
- ' section surface in the collection for the first cross section
- ' in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Dim xSectPCode As AeccCrossSectionPointCode

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

Set xSectPCode = xSect.CrossSectionPointCodes.Item(0)

```
' Get the alignment name
Dim alignName As String
alignName = align.Name
```

'Get the cross section station and format it
Dim station As String
station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The alignment name is: " & alignName & vbCrLf & _
"The first cross section is at station: " & station & vbCrLf & _
"The data for the first point code is: " & vbCrLf & _
vbTab & "Code: " & xSectPCode.Code & vbCrLf & _
vbTab & "Description: " & xSectPCode.Description & vbCrLf & _
vbTab & "Elevation: " & Format(xSectPCode.elevation, "0.00") & vbCrLf &
vbTab & "Offset: " & Format(xSectPCode.offset, "0.00"), _
vbInformation, "Description Example"

End Sub

Sub Example_Description_PointGroup()

- 'This function gets the Description for the first PointGroup
- ' in the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The Description for the first PointGroup is: " & pntGrp.Description, vbInformation, "Description Example"

End Sub

Sub Example_Description_Project()

- 'This example returns the Description setting for the first Project
- ' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Description value for the first project in the collection is: " & provbInformation, "Description Example"

End Sub

Sub Example_Description_Prototype()

'This example returns the Description for the first prototype in the collection.

Dim prot As AeccPrototype

Set prot = AeccApplication.Prototypes.Item(0)

MsgBox "The Description of the first prototype is " & prot.Description, _ vbInformation, "Description Example"

End Sub

Sub Example_Description_Surface()

'This example returns the Description for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The Description for the first surface is: " & surf.Description, _ vbInformation, "Description Example"

DescriptionFormat Example

Sub Example_DescriptionFormat()

- 'This example returns the DescriptionFormat for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey DescriptionFormat is " & dKey.Descrive vbInformation, "DescriptionFormat Example"

DescriptionKeyFiles Example

Sub Example_DescriptionKeyFiles()

'This example returns the number of DescriptionKeyFiles

' in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of DescriptionKeyFiles in the current Project is: " & pro , vbInformation, "DescriptionKeyFiles Example"

DescriptionLayer Example

Sub Example_DescriptionLayer()

- 'This example returns the DescriptionLayer for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey DescriptionLayer is " & dKey.Descrip vbInformation, "DescriptionLayer Example"

DescriptionOverride Example

Sub Example_DescriptionOverride()

```
'This function gets the DescriptionOverride for the first PointGroup
'in the collection.

Dim pntGrp As AeccPointGroup
Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

If pntGrp.DescriptionOverride = True Then
    MsgBox "The DescriptionOveride for the first PointGroup is turned on.", _
    vbInformation, "DescriptionOverride Example"

Else
    MsgBox "The DescriptionOveride for the first PointGroup is turned off.", _
    vbInformation, "DescriptionOverride Example"

End If
```

DescriptionXDRef Example

Sub Example_DescriptionXDRef()

'This function gets the DescriptionXDRef for the first PointGroup

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The DescriptionXDRef for the first PointGroup is: " & _ pntGrp.DescriptionXDRef, vbInformation, "DescriptionXDRef Example"

^{&#}x27; in the collection.

DifferenceGrid Example

Sub Example_DifferenceGrid()

```
'This example creates a surface from two existing, overlapping,
```

Dim surfs As AeccSurfaces

Dim diffSurf As AeccSurface

Set surfs = AeccApplication.ActiveProject.Surfaces

Dim surf0 As String

Dim surf1 As String

Dim surf2 As String

Dim gridx As Double

Dim gridy As Double

Dim pnt As Variant

Dim rows As Integer

Dim columns As Integer

' Get the difference grid input

surf0 = ThisDrawing.Utility.GetString(True, "Enter name for new Surface: ")

surf1 = ThisDrawing.Utility.GetString(True, "Enter name of first Surface: ")

surf2 = ThisDrawing.Utility.GetString(True, "Enter name of overlapping Surfa

gridx = ThisDrawing.Utility.GetReal("Enter the width of the grid: ")

gridy = ThisDrawing.Utility.GetReal("Enter the height of the grid: ")

rows = ThisDrawing.Utility.GetInteger("Enter the number of rows in the grid: columns = ThisDrawing.Utility.GetInteger("Enter the number of columns in th pnt = ThisDrawing.Utility.GetPoint(, "Select lower left point of grid: ")

' Create the difference grid

diffSurf = surfs.DifferenceGrid(surf0, surf1, surf2, gridx, gridy, rows, columns

MsgBox "The number of faces in the new surface is: " & diffSurf.NumberOfFa vbInformation, "DifferenceGrid Example"

^{&#}x27; surfaces using the Grid method.

Direction Example

Examples:

- 1 AlignTangent
- 1 ProfileBlock (Civil Engineering Feature)

```
Sub Example_Direction_AlignTangent()
```

```
'This example returns the Direction for the first Tangent found in the
```

```
' first Alignment in the collection.
```

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

```
Dim alignMsg As String
```

alignMsg = "There is no Tangent entity in the first Alignment."

```
' Find first Tangent in the alignment
```

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kTangent Then

alignMsg = "The Direction for the first Tangent in the alignment is: " & al

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "Direction Example"

End Sub

Sub Example_Direction_ProfileBlock()

- 'This example returns the Direction for the first ProfileBlock
- ' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The Direction for the first ProfileBlock in the collection is: " _ & alignProf.Direction, vbInformation, "Direction Example"

Documents Example

Sub Example_Documents()

'This example returns the number of drawings loaded

' in the current session.

Dim docs As AeccDocuments

Set docs = AeccApplication.Documents

MsgBox "The number of loaded drawings is: " & docs.Count, _ vbInformation, "Documents Example"

DoubleToStaFormat Example

Sub Example_DoubleToStaFormat()

' This example returns the DoubleToStaFormat string for 100.0. Dim aligns As AeccAlignments
Set aligns = AeccApplication.ActiveProject.Alignments

MsgBox "The DoubleToStaFormat for 100.0 is " & aligns.DoubleToStaFormat vbInformation, "DoubleToStaFormat Example"

DrainsInto Example

Sub Example_DrainsInto()

- 'This example returns the ID of the first WaterShed that is
- ' drained into for the first WaterShed in the collection.

Dim surf As AeccSurface

Dim wShed As AeccWaterShed

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wShed = surf.Outputs.WaterSheds.Item(0)

MsgBox "The first Watershed Id that is drained into is: " & wShed.Id _ , vbInformation, "DrainsInto Example"

DrawingPath Example

Sub Example_DrawingPath()

'This example returns the DrawingPath setting for the first Project

' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The DrawingPath value for the first Project in the collection is: " & I vbInformation, "DrawingPath Example"

Drawings Example

Sub Example_Drawings()

'This example returns the number of Drawings

' in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of Drawings in the current Project is: " & proj.Drawings , vbInformation, "Drawings Example"

DrawingSetupPath Example

Sub Example_DrawingSetupPath()

' This example returns the DrawingSetupPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for DrawingSetupPath is: " & prefFiles.DrawingSetupPath is: " & prefFiles.DrawingSetupPath Example"

Easting Example

Examples:

- 1 AeccPoint
- l CogoPoint
- 1 TinPoint

```
Sub Example_Easting_AeccPoint()
```

```
'This example returns the Easting setting for the
```

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

'Create the selection set based on a point selection

' and filter for the Point objects

Dim ssetObj As AcadSelectionSet
Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AECC_POINT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

^{&#}x27; first Point object in a selection set.

Dim objPoint As AeccPoint Set objPoint = ssetObj.Item(0)

MsgBox "The setting for Easting is: " & objPoint.Easting, vbInformation, "Eas

End Sub

Sub Example_Easting_CogoPoint()

'This example returns a Cogo point East coordinate.

Dim dbPoint As AeccCogoPoint

Set dbPoint = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The point Easting for the first point in the collection is: " & dbPoint." vbInformation, "Easting Example"

End Sub

Sub Example_Easting_TinPoint()

- 'This example returns the Easting of the first
- 'TIN Point in the collection.

Dim surf As AeccSurface

Dim tin As AeccTinPoint

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set tin = surf.Outputs.TinPoints.Item(0)

MsgBox "The Easting of the first TIN Point is: " & tin. Easting, vbInformation,

EastNorthToXy Example

```
Sub Example_EastNorthToXy()
```

```
'This example returns the X and Y for a Easting Northing.
Dim util As AeccUtility
Set util = AeccApplication.ActiveDocument.Utility
Dim var1 As Variant
Dim ptEN(0 To 2) As Double
Dim ptXY (0 To 2) As Double
Dim str As String
ptEN(0) = 1000
ptEN(1) = 1000
ptEN(2) = 0
'Convert point to X, Y
var1 = util.EastNorthToXy(ptEN)
ptXY(0) = var1(0)
ptXY(1) = var1(1)
ptXY(2) = var1(2)
' Make display string
str = Format(ptXY(0), "0.00") + ", " + Format(ptXY(1), "0.00") +
MsgBox "Point: " & str, vbInformation, "EastNorthXy Example"
```

Edges Example

Sub Example_Edges()

- 'This example returns the number of Edges in the first
- ' surface in the collection.

Dim surf As AeccSurface

Dim surfOut As AeccSurfaceOutputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfOut = surf.Outputs

MsgBox "The number of Edges in the first Surface is: " & surfOut.Edges.Cour , vbInformation, "Edges Example"

EGPrecision Example

Examples:

1 CrossSectionBlock (Civil Engineering Feature)

1 ProfileBlock (Civil Engineering Feature)

Sub Example_EGPrecision_CrossSectionBlock()

- 'This example returns the existing ground percision for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks
Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The existing ground precision for the alignment cross section at stati alignXSect.EGPrecision, vbInformation, "EGPrecision Example"

End Sub

Sub Example_EGPrecision_ProfileBlock()

- 'This example returns the EGPrecision for the first ProfileBlock
- ' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The EGPrecision for the first ProfileBlock in the collection is: " _ & alignProf.EGPrecision, vbInformation, "EGPrecision Example"

EGProfiles Example

Sub Example_EGProfiles()

- 'This example returns the number of existing ground profiles for the
- ' first alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The number of existing ground profiles in first alignment is: " $\&_$ align.EGProfiles.Count , vbInformation, "EGProfiles Example"

Elevation Example

Examples:

- l AeccContour
- l AeccPoint
- l CogoPoint
- 1 <u>CrossSectionPointCode</u> (Civil Engineering Feature)
- 1 ElevationContours
- l PointGroup
- 1 PVI (Civil Engineering Feature)
- 1 TinPoint

Sub Example_Elevation_AeccContour()

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- 'Create the selection set based on a point selection
- ' and filter for Contour objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0

^{&#}x27;This example displays The Elevation for a selected contour.

```
dataValue(0) = "AECC_CONTOUR"
  Dim groupCode As Variant
  Dim dataCode As Variant
  groupCode = gpCode
  dataCode = dataValue
  ssetObj.SelectOnScreen groupCode, dataCode
  'Get first contour from selection set
  Dim objContour As AeccContour
  Set objContour = ssetObj.Item(0)
  MsgBox "The Contour Elevation is: " & objContour. Elevation, vbInformation,
End Sub
Sub Example_Elevation_AeccPoint()
  'This example returns the Elevation setting for the
  ' first Point object in a selection set
  On Error Resume Next
  ' Delete existing SelectionSet
  ThisDrawing.SelectionSets("SSet").Delete
  'Create the selection set based on a point selection
  ' and filter for the Point objects
  Dim ssetObj As AcadSelectionSet
  Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
  Dim gpCode(0) As Integer
  Dim dataValue(0) As Variant
  gpCode(0) = 0
  dataValue(0) = "AECC POINT"
```

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objPoint As AeccPoint Set objPoint = ssetObj.Item(0)

MsgBox "The setting for Elevation is: " & objPoint.Elevation, vbInformation, '

End Sub

Sub Example_Elevation_CogoPoint()

'This example returns the Elevation for the first CogoPoint in the collection Dim cogoPnt As AeccCogoPoint Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Elevation for the first CogoPoint in the collection is: " & cogoPi

End Sub

Sub Example_Elevation_CrossSectionPointCode()

- 'This example returns the point code data for the first cross
- ' section surface in the collection for the first cross section
- ' in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Dim xSectPCode As AeccCrossSectionPointCode

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

```
Set xSectPCode = xSect.CrossSectionPointCodes.Item(0)
  ' Get the alignment name
  Dim alignName As String
  alignName = align.Name
  'Get the cross section station and format it
  Dim station As String
  station = aligns.DoubleToStaFormat(xSect.station)
  MsgBox "The alignment name is: " & alignName & vbCrLf & _
    "The first cross section is at station: " & station & vbCrLf & _
    "The data for the first point code is: " & vbCrLf & _
    vbTab & "Code: " & xSectPCode.Code & vbCrLf &
    vbTab & "Description: " & xSectPCode.Description & vbCrLf &
    vbTab & "Elevation: " & Format(xSectPCode.elevation, "0.00") & vbCrLf &
    vbTab & "Offset: " & Format(xSectPCode.offset, "0.00"), _
    vbInformation, "Elevation Example"
End Sub
Sub Example_Elevation_ElevationContours()
  'This example returns the number of ElevationContours at 100.0 for
  ' the first surface in the collection
  Dim surf As AeccSurface
  Dim elevContours As AeccElevationContours
  Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Set elevContours = Surf.Outputs.ElevationContours
  elevContours.Elevation = 100#
  MsgBox "The number of ElevationContours at 100.0 is: " & elevContours.Cou
```

End Sub

, vbInformation, "Elevation Example"

Sub Example_Elevation_PointGroup()

- 'This function gets the Elevation for the first PointGroup
- ' in the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The Elevation for the first PointGroup is: " & _ pntGrp.Elevation, vbInformation, "Elevation Example"

End Sub

Sub Example_Elevation_PVI()

- 'This example returns the Elevation value for the second PVI in the
- ' first finished ground of the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Dim PVI As AeccPVI

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

Set PVI = FGProf.PVIs.Item(1)

MsgBox "The Elevation of the second PVI is: " _ & Format(PVI.Elevation, "0.000"), vbInformation, "Elevation Example"

End Sub

Sub Example_Elevation_TinPoint()

- ' This example returns the Elevation of the first
- 'TIN Point in the collection.

Dim surf As AeccSurface

Dim tin As AeccTinPoint

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set tin = surf.Outputs.TinPoints.Item(0)

MsgBox "The Elevation of the first TIN Point is: " & tin.Elevation, _ vbInformation, "Elevation Example"

ElevationAt Example

Examples:

1 <u>EGProfile</u> (Civil Engineering Feature)

1 FGProfile (Civil Engineering Feature)

Sub Example_ElevationAt_EGProfile()

'This example uses ElevationAt to displays the elevation for a station,

' for the first existing ground profile in the first alignment in the collection.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

Dim station As Double

Dim elevation As Double

'Get the station and the elevation

station = ThisDrawing.Utility.GetReal("Enter a station: ")
elevation = EGProf.ElevationAt(station)

MsgBox "The elevation for station " & Format(station, "0.00") & " is: " _ & Format(elevation, "0.00"), vbInformation, "ElevationAt Example" End Sub

Sub Example_ElevationAt_FGProfile()

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

^{&#}x27;This example uses ElevationAt to displays the elevation for a station,

^{&#}x27; for the finished ground profile in the first alignment in the collection.

Dim station As Double
Dim elevation As Double

'Get the station and the elevation station = ThisDrawing.Utility.GetReal("Enter a station: ") elevation = FGProf.ElevationAt(station)

MsgBox "The elevation for station " & Format(station, "0.00") & " is: " _ & Format(elevation, "0.000"), vbInformation, "ElevationAt Example" End Sub

ElevationContours Example

Sub Example_ElevationContours()

- 'This example returns the number of ElevationContours at 100.0,
- ' in the first surface in the collection.

Dim surf As AeccSurface

Dim surfOut As AeccSurfaceOutputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfOut = surf.Outputs

surfOut.ElevationContours.Elevation = 100#

MsgBox "The number of ElevationContours at 100.0 in the first Surface is: " 8 surfOut.ElevationContours.Count, vbInformation, "ElevationContours Exan

ElevationOverride Example

Sub Example_ElevationOverride()

ElevationPrecision Example

Sub Example_ElevationPrecision()

'This example returns the ElevationPrecision setting for the current drawing. Dim dbPref As AeccDatabasePreferences Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for ElevationPrecision is: " & dbPref.ElevationPre vbInformation, "ElevationPrecision Example"

ElevationXDRef Example

Sub Example_ElevationXDRef()

'This function gets the ElevationXDRef for the first PointGroup

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The ElevationXDRef for the first PointGroup is: " & _ pntGrp.ElevationXDRef, vbInformation, "ElevationXDRef Example"

^{&#}x27; in the collection.

EndDirection Example

Sub Example_EndDirection()

```
'This example returns the EndDirection for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment
Dim alignEnts As AeccAlignEntity
Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment
For Each alignEnt In align.AlignEntities
If alignEnt.Type = kCurve Then
alignMsg = "The EndDirection for the first Curve in the alignment is: " & Exit For End If
Next

MsgBox alignMsg, vbInformation, "EndDirection Example"
```

EndEasting Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 ParcelEntity, ParcelCurve, ParcelLine

Sub Example_EndEasting_AlignEntity()

- 'This example returns the EndEasting for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The EndEasting of the first entity in the Alignment is: " & alignEnt.I vbInformation, "EndEasting Example"

End Sub

Sub Example_EndEasting_ParcelEntity()

- 'This example returns the EndEasting for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The EndEasting of the first entity in the Parcel is: " & parcelEnt.End vbInformation, "EndEasting Example"

EndingStation Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 Alignment
- 1 ProfileBlock (Civil Engineering Feature)

Sub Example_EndingStation_AlignEntity()

- 'This example returns the EndingStation for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The EndingStation of the first entity in the Alignment is: " & _ alignEnt.EndingStation, vbInformation, "EndingStation Example"

End Sub

Sub Example_EndingStation_Alignment()

'This example returns the EndingStation for the first Alignment in the collectic Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The EndingStation of the first Alignment is: " & align.EndingStation vbInformation, "EndingStation Example"

End Sub

Sub Example_EndingStaion_ProfileBlock()

'This example returns the EndingStation for the first ProfileBlock

Dim alignProf As AeccProfileBlock
Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The EndingStation for the first ProfileBlock in the collection is: " _ & alignProf.EndingStation, vbInformation, "EndingStation Example"

^{&#}x27; in the collection

EndNorthing Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 ParcelEntity, ParcelCurve, ParcelLine

Sub Example_EndNorthing_AlignEntity()

- 'This example returns the EndNorthing for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The EndNorthing of the first entity in the Alignment is: " & alignEnt vbInformation, "EndNorthing Example"

End Sub

Sub Example_EndNorthing_ParcelEntity()

- 'This example returns the EndNorthing for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The EndNorthing of the first entity in the Parcel is: " & parcelEnt.En vbInformation, "EndNorthing Example"

ExceedBoth Example

Sub Example_ExceedBoth()

'This example returns the ExceedBoth setting for the WaterSheds.

Dim surf As AeccSurface

Dim wSheds As AeccWaterSheds

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wSheds = surf.Outputs.WaterSheds

MsgBox "The ExceedBoth setting for WaterSheds is: " & wSheds.ExceedBoth vbInformation, "ExceedBoth Example"

ExtEasting Example

```
Sub Example_ExtEasting()
```

```
'This example returns the ExtEasting for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The ExtEasting for the first Spiral in the alignment is: " & aliexit For
End If

Next

MsgBox alignMsg, vbInformation, "ExtEasting Example"
```

ExternalSecant Example

Sub Example_ExternalSecant()

```
'This example returns the ExternalSecant for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The ExternalSecant for the first Curve in the alignment is: "
Exit For
End If

Next

MsgBox alignMsg, vbInformation, "ExternalSecant Example"
```

ExternalStaToInternal Example

Sub Example_ExternalStaToInternal()

'This example gets the ExternalStaToInternal of 100.0

' for the first Alignment in the collection..

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim intSta As Variant

intSta = align.ExternalStaToInternal(100#)

MsgBox "The ExternalStaToInternal of 100.0 for the first Alignment is: " & in vbInformation, "ExternalStaToInternal Example"

ExtNorthing Example

Sub Example_ExtNorthing()

```
'This example returns the ExtNorthing for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment
Dim alignEnts As AeccAlignEntity
Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment
For Each alignEnt In align.AlignEntities
If alignEnt.Type = kSpiral Then
alignMsg = "The ExtNorthing for the first Spiral in the alignment is: " & a
Exit For
End If
Next

MsgBox alignMsg, vbInformation, "ExtNorthing Example"
```

Faces Example

Sub Example_Faces()

- 'This example returns the number of Faces in the first
- ' surface in the collection.

Dim surf As AeccSurface

Dim surfOut As AeccSurfaceOutputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfOut = surf.Outputs

MsgBox "The number of Faces in the first Surface is: " & surfOut.Faces.Count, vbInformation, "Faces Example"

FacetDeviation Example

Sub Example_FacetDeviation()

'This example displays the FacetDeviation setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The setting for FacetDeviation is: " & dbPref.FacetDeviation, _ vbInformation, "FacetDeviation Example"

FGPrecision Example

Examples:

1 <u>CrossSectionBlock</u> (Civil Engineering Feature)

1 ProfileBlock (Civil Engineering Feature)

Sub Example_FGPrecision_CrossSectionBlock()

- 'This example returns the finished ground percision for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks

Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The finished ground precision for the alignment cross section at stati alignXSect.FGPrecision, vbInformation, "FGPrecision Example"

End Sub

Sub Example_FGPrecision_ProfileBlock()

- 'This example returns the FGPrecision for the first ProfileBlock
- ' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The FGPrecision for the first ProfileBlock in the collection is: " _ & alignProf.FGPrecision, vbInformation, "FGPrecision Example"

FGProfiles Example

Sub Example_FGProfiles()

- 'This example returns the number of finished ground pofiles for the
- ' first alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The number of finished groung profiles in first alignment is: " _ & align.FGProfiles.Count , vbInformation, "FGProfiles Example"

File Example

Sub Example_File()

'This example returns the File for the first FileLock in the collection.

Dim filelock As AeccFileLock

Set filelock = AeccApplication.ActiveProject.fileLocks.Item(0)

MsgBox "The File of the first FileLock in the collection is: " & filelock.File, _ vbInformation, "File Example"

FileLocks Example

Sub Example_FileLocks()

'This example returns the number of FileLocks

' in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of FileLocks in the current Project is: " & proj.FileLock , vbInformation, "FileLocks Example"

Files Example

Sub Example_Files()

- 'This example returns the current setting of PrototypePath
- ' from the preferences object.

Dim prefs As AeccPreferences

Set prefs = AeccApplication.Preferences

MsgBox "The current value for PrototypePath is: " & prefs.Files.PrototypePath vbInformation, "Files Example"

FindAllConnectingEdges Example

Sub Example_FindAllConnectingEdges()

```
'This example returns all the connecting edges for a given point
' in the first surface in the surface collection
Dim surf As AeccSurface
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
'Get point for calculating connecting edges
Dim pnt As Variant
pnt = ThisDrawing.Utility.GetPoint(, "Select point in the surface: ")
'Get an actual surface point.
pnt = surf.FindPoint(pnt(0), pnt(1))
'Get edge points
Dim edgePnts As Variant
edgePnts = surf.FindAllConnectingEdges(pnt(0), pnt(1))
Dim lineObj As AcadLine
Dim pnt1(0 To 2) As Double
Dim pnt2(0 To 2) As Double
Dim pnt3(0 To 2) As Double
' Draw red lines around first triangle definition
pnt1(0) = edgePnts(0)
pnt1(1) = edgePnts(1)
pnt1(2) = edgePnts(2)
pnt2(0) = edgePnts(3)
pnt2(1) = edgePnts(4)
pnt2(2) = edgePnts(5)
pnt3(0) = edgePnts(6)
pnt3(1) = edgePnts(7)
pnt3(2) = edgePnts(8)
Set lineObj = ThisDrawing.ModelSpace.AddLine(pnt1, pnt2)
```

```
lineObj.Color = acRed
Set lineObj = ThisDrawing.ModelSpace.AddLine(pnt2, pnt3)
lineObj.Color = acRed
Set lineObj = ThisDrawing.ModelSpace.AddLine(pnt3, pnt1)
lineObj.Color = acRed
```

FindAllFaces Example

Sub Example_FindAllFaces()

```
'This example begins by getting a faces collection for an entered
```

Dim surf As AeccSurface

Dim faces As AeccFaces

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set faces = surf.Outputs.faces

'Return a point using a prompt

Dim returnPnt As Variant

returnPnt = ThisDrawing.Utility.GetPoint(, "Select a point: ")

' Find face points

faces.FindFace returnPnt(0), returnPnt(1)

MsgBox "The number of Faces in the collection after FindFace is: " & faces.Complete vbInformation, "FindAllFaces Example"

faces.FindAllFaces

MsgBox "The number of Faces in the collection after FindAllFaces is: " & face vbInformation, "FindAllFaces Example"

^{&#}x27; point in the first surface in the collection. Then the FindAllFaces

^{&#}x27; is used to get the entire faces collection.

FindConnectingEdges Example

Sub Example_FindConnectingEdge()

```
'This example returns the connecting edge for a given point
' in the first surface in the collection.
Dim surf As AeccSurface
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
'Get point for calculating connecting edges
Dim pnt As Variant
pnt = ThisDrawing.Utility.GetPoint(, "Select point in the surface: ")
'Get edge points
Dim edgePnts As Variant
edgePnts = surf.FindConnectingEdge(pnt(0), pnt(1))
Dim lineObj As AcadLine
Dim startPnt(0 To 2) As Double
Dim endPnt(0 To 2) As Double
' Draw a red line along edge
startPnt(0) = edgePnts(0)
startPnt(1) = edgePnts(1)
startPnt(2) = edgePnts(2)
endPnt(0) = edgePnts(3)
endPnt(1) = edgePnts(4)
endPnt(2) = edgePnts(5)
Set lineObj = ThisDrawing.ModelSpace.AddLine(startPnt, endPnt)
lineObj.Color = acRed
```

FindFace Example

Sub Example_FindFace()

```
'This example returns a new face collection for an entered point in
```

Dim surf As AeccSurface

Dim faces As AeccFaces

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set faces = Surf.Outputs.Faces

MsgBox "The number of Faces in the collection before FindFace is: " & faces.", vbInformation, "FindFace Example"

'Return a point using a prompt

Dim returnPnt As Variant

returnPnt = ThisDrawing.Utility.GetPoint(, "Select a point: ")

' Find face points

faces.FindFace returnPnt(0), returnPnt(1)

MsgBox "The number of Faces in the collection after FindFace is: " & faces.Co., vbInformation, "FindFace Example"

^{&#}x27; the first surface in the collection.

FindPath Example

Sub Example_FindPath()

```
'This example returns the face collection for an entered path in
```

Dim surf As AeccSurface

Dim faces As AeccFaces

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set faces = surf.Outputs.faces

MsgBox "The number of Faces in the collection before FindPath is: " & faces.", vbInformation, "FindPath Example"

'Select the path for faces

Dim pnt1 As Variant

Dim pnt2 As Variant

pnt1 = ThisDrawing.Utility.GetPoint(, "Select first point on a surface: ")

pnt2 = ThisDrawing.Utility.GetPoint(pnt1, "Select second point on a surface: "

faces.FindPath pnt1(0), pnt1(1), pnt2(0), pnt2(1)

MsgBox "The number of Faces in the collection before FindPath is: " & faces.", vbInformation, "FindPath Example"

^{&#}x27; the first surface in the collection.

FindPoint Example

Sub Example_FindPoint()

```
'This example draws a point at the surface point for a selected point
' in the first surface in the collection. The point id drawn with the default
' point style.
Dim surf As AeccSurface
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
'Get point for calculating point
Dim pnt As Variant
pnt = ThisDrawing.Utility.GetPoint(, "Select point in the surface: ")
' Get surface point
Dim edgePnts As Variant
edgePnts = surf.FindPoint(pnt(0), pnt(1))
Dim pntObj As AcadPoint
Dim startPnt(0 To 2) As Double
'Draw a red point at found point
startPnt(0) = edgePnts(0)
startPnt(1) = edgePnts(1)
startPnt(2) = edgePnts(2)
Set pntObj = ThisDrawing.ModelSpace.AddPoint(startPnt)
pntObj.Color = acRed
```

FirstTimeDrawingSetup Example

Sub Example_FirstTimeDrawingSetup()

```
'This example returns the FirstTimeDrawingSetup setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User
'Convert the constant to a string.
Dim strType As String
If prefUser.FirstTimeDrawingSetup = kDrawingSetupWizard Then
  strType = "Wizard"
End If
If prefUser.FirstTimeDrawingSetup = kDrawingSetupCommand Then
  strType = "Command"
End If
If prefUser.FirstTimeDrawingSetup = kAutoloadSetupFile Then
  strType = "Autoload"
End If
MsgBox "The current value for FirstTimeDrawingSetup is: " & strType, _
  vbInformation, "FirstTimeDrawingSetup Example"
```

FirstTimeDrawingSetupFile Example

Sub Example_FirstTimeDrawingSetupFile()

'This example returns the FirstTimeDrawingSetupFile setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for FirstTimeDrawingSetupFile is: " & _ prefUser.FirstTimeDrawingSetupFile, vbInformation, "FirstTimeDrawingSe

Format Example

Sub Example_Format()

- 'This example returns the Format setting for the first PointFile
- ' in the collection.

Dim surf As AeccSurface

Dim pntFile As AeccPointFile

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set pntFile = surf.Inputs.PointFiles.Item(0)

MsgBox "The Format for the first PointFile in the collection is: " & pntFile.Fo , vbInformation, "Format Example"

FormatsPath Example

Sub Example_FormatsPath()

' This example returns the FormatsPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for FormatsPath is: " & prefFiles.FormatsPath, _ vbInformation, "FormatsPath Example"

FullDescription Example

Sub Example_FullDescription()

' This example returns the FullDescription for the first CogoPoint in the collect ${\bf Dim\ cogoPnt\ As\ AeccCogoPoint}$

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The FullDescription for the first CogoPoint in the collection is: " & _ cogoPnt.FullDescription, vbInformation, "FullDescription Example"

FullName Example

Examples:

- 1 DescriptionKeyFile
- 1 Drawing
- l Project

Sub Example_FullName_DescriptionKeyFile()

- 'This example returns the FullName for the first
- 'DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

MsgBox "The value for DescriptionKeyFile FullName is " & dKeyFile.FullNa, vbInformation, "FullName Example"

End Sub

Sub Example_FullName_Drawing()

- 'This example returns the FullName setting for the first Drawing
- ' in the collection

Dim dwg As AeccDrawing

Set dwg = AeccApplication.ActiveProject.Drawings(0)

MsgBox "The FullName for the first Drawing in the collection is: " & dwg.Ful , vbInformation, "FullName Example"

End Sub

Sub Example_FullName_Project()

- 'This example returns the FullName setting for the first Project
- ' in the collection

Dim proj As AeccProject Set proj = AeccApplication.Projects.Item(0)

MsgBox "The FullName value for the first Project in the collection is: " & proj vbInformation, "FullName Example"

GetBoundingBox Example

Sub Example_GetBoundingBox()

- 'This example returns the two points defining the Bounding Box
- ' for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Dim pntLL As Variant
Dim pntUR As Variant

'Build the surface and get the bounding box points surf.Build surf.GetBoundingBox pntLL, pntUR

MsgBox "Lower left bounding box point: " & pntLL(0) & ", " & pntLL(1) & " vbCrLf & _

"Upper right bounding box point: " & pntUR(0) & ", " & pntUR(1) & ", " & , vbInformation, "GetBoundingBox Example"

GetDouble Example

Sub Example_GetDouble()

- 'This example uses GetDouble to return the ContLabelSpacingDist
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

MsgBox "The value for ContLabelSpacingDist is " & surfPref.GetDouble(kCo vbInformation, "GetDouble Example"

GetElevation Example

Sub Example_GetElevation()

```
'This example returns the Elevation for an entered point for the first
```

Dim surfs As AeccSurfaces

Dim surf As AeccSurface

Set surfs = AeccApplication.ActiveProject.Surfaces

Set surf = surfs.Item(0)

Dim returnPnt As Variant

Dim elev As Double

' Get a point

returnPnt = ThisDrawing.Utility.GetPoint(, "Select a point: ")

' Get the elevation from the selected point elev = surf.GetElevation(returnPnt(0), returnPnt(1))

MsgBox "The elevation from GetElevation is " & elev, vbInformation, "GetEle

^{&#}x27; surface in the collection.

Getinteger Example

Sub Example_GetInteger()

- 'This example uses GetInteger to return the ContLabelPrecision
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

MsgBox "The value for ContLabelPrecision is " & surfPref.GetInteger(kContL vbInformation, "GetInterger Example"

GetLayerName Example

Examples:

1 CrossSectionBlock (Civil Engineering Feature)

1 ProfileBlock (Civil Engineering Feature)

Sub Example_GetLayerName_CrossSectionBlock()

- 'This example uses GetLayerName to get the grid layer name for the
- ' first cross section in the first alignment profile in the collection.

Dim alignXSect As AeccCrossSectionBlock

Set alignXSect = AeccApplication.ActiveDocument.CrossSectionBlocks.Item(

MsgBox "The exisiting ground layer for the first alignment cross section in the & alignXSect.GetLayerName(kCrossSectionEGLayer), vbInformation, "Ge

End Sub

Sub Example_GetLayerName_ProfileBlock()

- 'This example uses GetLayerName to get the grid layer name for the
- ' first alignment profile in the collection.

Dim alignProf As AeccAlignProfile

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The grid layer for the first ProfileBlock in the collection is: "_ & alignProf.GetLayerName(kGrid), vbInformation, "GetLayerName Example Examp

GetStaStrWithEquations Example

Sub Example_GetStaStrWithEquation()

'This example gets the GetStaStrWithEquations from a user supplied station

' for the first Alignment in the collection.

Dim Align As AeccAlignment

Set Align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim staEqu As String

Dim Station As Double

' Get the station

Station = ThisDrawing.Utility.GetReal("Enter the station: ")

'Get the station with Station Equations applied staEqu = Align.GetStaStrWithEquations(Station)

MsgBox "The GetStaStrWithEquation value for station " & Station & " is: " &

GetStaWithEquation Example

Sub Example_GetStaWithEquation()

```
'This example gets the GetStaWithEquations from a user supplied station
```

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim staEqu As Double

Dim station As Double

' Get the station

station = ThisDrawing.Utility.GetReal("Enter the station: ")

' Get the station with Station Equations applied staEqu = align.GetStaWithEquations(station)

MsgBox "The GetStaWithEquation value for station " & station & " is: " & sta

^{&#}x27; for the first Alignment in the collection.

GetString Example

Sub Example_GetSting()

- 'This example uses GetString to return the SurfaceLayer
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

 $MsgBox "The \ value \ for \ Surface Layer \ is " \& surf Pref. Get String (k Surface Layer \ and the surface Layer) when the surface of the$

GridEasting Example

Sub Example_GridEasting()

'This example returns the GridEasting for the first CogoPoint in the collection.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The GridEasting for the first CogoPoint in the collection is: " & cogo vbInformation, "GridEasting Example"

GridNorthing Example

Sub Example_GridNorthing()

' This example returns the GridNorthing for the first CogoPoint in the collectio Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The GridNorthing for the first CogoPoint in the collection is: " & coq vbInformation, "GridNorthing Example"

GroupName Example

Examples:

l CogoPoint

1 PointGroup

Sub Example_GroupName_CogoPoint()

- 'This example returns the GroupName for the first CogoPoint in the collection
- 'This example assumes the first point in the collection is in a
- ' Point Group named "Example Group".

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

'Filter CogoPoints collection on GroupName cogoPnts.GroupName = "Example Group"

Set cogoPnt = cogoPnts.Item(0)

MsgBox "The GroupName for the first CogoPoint in the collection is: " & cogo vbInformation, "GroupName Example"

End Sub

Sub Example_GroupName_PointGroup()

- 'This function gets the GroupName for the first PointGroup
- ' in the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The GroupName for the first PointGroup is: " & _ pntGrp.GroupName, vbInformation, "GroupName Example"

Height Example

Sub Example_Height()

- 'This example returns the height for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

 ${\bf Set\ align XSects} = {\bf Aecc Application. Active Document. Cross Section Blocks}$

Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The height for the alignment cross section at station " & station & " i Format(alignXSect.Height, "0.00"), vbInformation, "Height Example"

HelpPath Example

Sub Example_HelpPath()

' This example returns the HelpPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for HelpPath is: " & prefFiles.HelpPath, _ vbInformation, "HelpPath Example"

ID Example

Examples:

- 1 **Boundary**
- 1 Breakline
- 1 ContourItem
- 1 WaterShed

Sub Example_Id_Boundary()

```
'This example returns the Id
```

' for the first Boundary in the collection.

Dim surf As AeccSurface

Dim bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bound = surf.Inputs.Boundaries.Item(0)

MsgBox "The Id for the Boundary is: " & bound.Id _ , vbInformation, "Id Example"

End Sub

Sub Example_Id_BreakLine()

```
'This example returns the Id
```

Dim surf As AeccSurface

Dim brkLine As AeccBreakLine

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set brkLine = surf.Inputs.BreakLines.Item(0)

```
MsgBox "The Id for the BreakLine is: " & brkLine.Id _ , vbInformation, "Id Example"
```

^{&#}x27; for the first BreakLine in the collection.

End Sub

Sub Example_Id_ContourItem()

```
'This example returns the Id
```

' for the first ContourItem in the collection.

Dim surf As AeccSurface

Dim cont As AeccContourItem

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set cont = Surf.Inputs.ContourItems.Item(0)

MsgBox "The Id for the ContourItem is: " & cont.Id _ , vbInformation, "Id Example"

End Sub

Sub Example_Id_Watershed()

'This example returns the ID of the first WaterShed in the collection.

Dim surf As AeccSurface

Dim wShed As AeccWaterShed

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wShed = surf.Outputs.WaterSheds.Item(0)

MsgBox "The Id for the first WaterShed in the collection is: " & wShed.Id _ , vbInformation, "Id Example"

Import Example

Examples:

- 1 Alignment
- 1 CrossSection (Civil Engineering Feature)
- 1 <u>EGProfile</u> (Civil Engineering Feature)
- 1 Parcel
- 1 Surface

Sub Example_Import_Alignment()

'This example adds an Alignment made up of a tangent.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

- ' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)
- ' Add a tangent

Dim tangent As AeccAlignTangent

Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

' Import the alignment to the drawing align.Import

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti vbInformation, "Import Example"

End Sub

Sub Example_Import_CrossSection()

'This example inserts the first cross section in the

' cross section collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

'Get the insertion point for the cross section

Dim returnPnt As Variant

returnPnt = ThisDrawing.Utility.GetPoint _

(, "Select bottom insertion point for the cross setion at station " & station & '

'Import cross section with complete geometry xSect.import returnPnt, False

End Sub

Sub Example_Import_EGProfile()

- 'This example imports the the first existing ground profile
- ' in the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set EGProf = align.EGProfiles.Item(0)

'Set the name of the first alignment to the current alignment aligns.CurrentAlignment = align.Name

```
'Determine default starting and ending station
Dim min As Double
Dim asta As Variant
asta = EGProf.StationElevations
min = asta(1)
Dim i As Integer
For i = 0 To UBound(asta) - 1 Step 2
  If asta(i + 1) < min Then
    min = asta(i + 1)
  End If
Next
'Get the start point for the existing ground profile
Dim returnPnt As Variant
returnPnt = ThisDrawing.Utility.GetPoint(, "Select a starting point:")
On Error Resume Next
'Get gtating station
Dim startSta As Double
startSta = asta(0)
startSta = ThisDrawing.Utility.GetReal("Starting Station <" & startSta & ">: ")
'Get ending station
Dim endSta As Double
endSta = asta(UBound(asta) - 1)
endSta = ThisDrawing.Utility.GetReal("Ending Station: <" & endSta & ">: ")
'Get datum elevation
Dim datumElev As Double
datumElev = min
datumElev = ThisDrawing.Utility.GetReal("Datum Elevation: <" & datumElev
'Get vertical scale
Dim vertScale As Double
```

```
Dim pref As AeccDatabasePreferences
  Set pref = AeccApplication.ActiveDocument.Preferences
  vertScale = pref.VerticalScale
  vertScale = ThisDrawing.Utility.GetReal("Vertical Scale: <" & vertScale & ">:
  'Get direction
  Dim dir As Boolean
  dir = True
  Dim ans As String
  ans = ThisDrawing.Utility.GetString(0, "Direction LeftToRight (Y/N): ")
  dir = Switch(ans = "Y", True, ans = "y", True, ans = "N", False, ans = "n", False
  'Get block only
  Dim block As Boolean
  block = True
  Dim ans As String
  ans = ThisDrawing.Utility.GetString(0, "Block only (Y/N): ")
  block = Switch(ans = "Y", True, ans = "y", True, ans = "N", False, ans = "n", F
  'Import the existing ground profile
  EGProf.import returnPnt, startSta, endSta, datumElev, vertScale, dir, block
End Sub
Sub Example_Import_Parcel()
  'This example adds a Parcel made up of lines and an arc.
  'The parcel is then imported into the drawing.
  Dim parcels As AeccParcels
  Dim parcel As AeccParcel
  Set parcels = AeccApplication.ActiveProject.parcels
  ' Add a new Parcel
  Set parcel = parcels.Add("New Parcel")
  ' Add lines and a curve to the Parcel
  parcel.AddLine 50#, 50#, 150#, 50#
```

parcel.AddLine 150#, 50#, 150#, 200# parcel.AddCurve 150#, 200#, 100#, 200#, 50#, 200#, True parcel.AddLine 50#, 200#, 50#, 50#

' Import the new Parcel parcel.Import

MsgBox "The total number of entities in the parcel is: " & parcel.ParcelEntities vbInformation, "Import Example"

End Sub

Sub Example_Import_Surface()

'This example draws the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

surf.Import

MsgBox "The Name of the first surface just imported is: " & surf.Name, _ vbInformation, "Import Example"

Inputs Example

Sub Example_Inputs()

- 'This example returns the Count of Pointfiles in the first Surface
- ' in the collection by using the Inputs property.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The number of PointFiles in the first surface is: " & surf.Inputs.Point , vbInformation, "Inputs Example"

InstantGrade Example

Examples:

l <u>EGProfile</u> (Civil Engineering Feature)

1 FGProfile (Civil Engineering Feature)

Sub Example_InstantGrade_FGProfile()

'This example gets a station from the user and returns the grade for

' the first existing ground profile in the first alignment in the collection.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

Dim station As Double

Dim grade As Double

Dim prompt As String

'Format the prompt and get the station

prompt = "Enter a station on alignment " & align.Name & ": "
station = ThisDrawing.Utility.GetReal(prompt)

'Get the grade

EGProf.InstantGrade station, grade

MsgBox "The grade at station and format the prompt" & Format(station, "0.00 & Format(grade, "0.00") & "%.", vbInformation, "InstantGrade Example"

End Sub

Sub Example_InstantGrade_FGProfile()

'This example gets a station from the user and returns the grade and difference

```
' for the first finished ground profile in the first alignment in the collection.
Dim align As AeccAlignment
Dim FGProf As AeccFGProfile
Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Set FGProf = align.FGProfiles.Item(0)
Dim station As Double
Dim grade As Double
Dim diff As Double
Dim prompt As String
'Format the prompt and get the station
prompt = "Enter a station on alignment " & align.Name & ": "
station = ThisDrawing.Utility.GetReal(prompt)
'Get the grade and difference
FGProf.InstantGrade station, grade, diff
MsgBox "The grade at station " & Format(station, "0.00") & " is: " _
  & Format(grade, "0.00") & vbCrLf & _
  "% and the difference is " & Format(diff, "0.00") & "%.", _
  vbInformation, "InstantGrade Example"
```

IsBreakline Example

Sub Example_IsBreakline()

- 'This example returns the IsBreakLine setting
- ' for the first Boundary in the collection.

Dim surf As AeccSurface

Dim bound As AeccBoundary

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set bound = surf.Inputs.Boundaries.Item(0)

MsgBox "The IsBreakline setting for the Boundary is: " & bound.IsBreakLine , vbInformation, "IsBreakLine Example"

IsNameSupported Example

Sub Example_IsNameSupported()

' This gets the IsNameSupported for the CogoPoints collection Dim cogoPnts As AeccCogoPoints Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

MsgBox "The IsNameSupported for CogoPoints is: " & cogoPnts.IsNameSupported Example"

IsVisible Example

Sub Example_IsVisible()

```
'This example returns the visibility setting for the
```

Dim surf As AeccSurface

Dim face As AeccFace

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set face = surf.Outputs.Faces.Item(0)

' Checks if the Face is visible

If face.IsVisible = True Then

MsgBox "The first Face in the collection is Visible!", vbInformation, "IsVisi Else

MsgBox "The first Face in the collection is Invisible!", vbInformation, "IsVi End If

^{&#}x27; first face item in the collection.

Item Example

Sub Example_Item()

- 'This example show two uses of the Item method.
- 'The first uses Item with an index counter to return an item in a collection.
- 'The second uses Item with a string to return an item in a collection.
- ' Iterate through the model space collection,
- ' get all the items in the collection
- ' and store them in an array called newObjs.

Dim count As Integer

count = AeccApplication.Projects.count

ReDim newObjs(count) As AeccProjects

Dim index As Integer

For index = 0 To count - 1

Set newObjs(index) = AeccApplication.Projects.Item(index)

Next

'Get a particular item, in this case a project, based on name "shaker".

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item("shaker")

K Example

```
Sub Example_K()
```

```
'This example returns the K value for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The K value for the first Spiral in the alignment is: " & align Exit For End If

Next

MsgBox alignMsg, vbInformation, "K Example"
```

Keywords Example

Sub Example_Keywords()

'This example returns the Keywords setting for the first Project

' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Keywords value for the first Project in the collection is: " & provbInformation, "Keywords Example"

Label Example

Sub Example_Label()

'This example returns the Label for the first FileLock in the collection.

Dim filelock As AeccFileLock

Set filelock = AeccApplication.ActiveProject.fileLocks.Item(0)

MsgBox "The Label of the first FileLock in the collection is: " & filelock.Labe vbInformation, "Label Example"

LabelPoints Example

coords = objContour.LabelPoints

```
Sub Example_LabelPoints()
  'This example displays a LabelPoint for a selected contour.
  On Error Resume Next
  ' Delete existing SelectionSet
  ThisDrawing.SelectionSets("SSet").Delete
  ' Create the selection set based on a point selection
  ' and filter for Contour objects
  Dim ssetObj As AcadSelectionSet
  Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
  Dim mode As Integer
  Dim gpCode(0) As Integer
  Dim dataValue(0) As Variant
  gpCode(0) = 0
  dataValue(0) = "AECC_CONTOUR"
  Dim groupCode As Variant
  Dim dataCode As Variant
  Dim returnPnt As Variant
  groupCode = gpCode
  dataCode = dataValue
  returnPnt = ThisDrawing.Utility.GetPoint(, "Select a contour line: ")
  ssetObj.SelectAtPoint returnPnt, groupCode, dataCode
  Dim objContour As AeccContour
  Set objContour = ssetObj.Item(0)
  Dim coords As Variant
```

 $\label{lem:matter} MsgBox "The first point in LabelPoints is " \& coords(0) \& ", " \& coords(0) \& vbInformation, "LabelPoints Example"$

LabelStyle Example

```
Sub Example_LabelStyle()
```

```
'This function gets the label style override for the
```

Dim ptGrp As AeccPointGroup

Set ptGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

' Initialize message to no label style overide defined

Dim labelMsg As String

labelMsg = "There is no label style override for point group " & _
ptGrp.GroupName & "."

'Test if label style is defined and if true define new message If ptGrp.LabelStyle ""Then

labelMsg = "The label style override for point group " & _
 ptGrp.GroupName & " is: " & ptGrp.LabelStyle
End If

MsgBox labelMsg, vbInformation, "LabelStyle Example"

^{&#}x27; project database point group

LabelStyleOverride Example

Sub Example_LabelStyleOverride()

'This function determines if the point group label style is overridden. Dim ptGrp As AeccPointGroup

Set ptGrp = AeccApplication. ActiveProject. PointGroups. Item(0)

MsgBox "Label style override for the project database point group is set to: " & ptGrp.LabelStyleOverride, vbInformation, "LabelStyleOverride Example"

LabelStylePath Example

Sub Example_LabelStylePath()

' This example returns the LabelStylePath setting. Dim prefFiles As AeccPreferencesFiles Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for LabelStylePath is: " & prefFiles.LabelStylePath vbInformation, "LabelStylePath Example"

LabelStyleXDRef Example

Sub Example_LabelStyleXDRef()

```
'This function gets the label style XDRef override for the
```

Dim ptGrp As AeccPointGroup

Set ptGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

' Initialize message to no label style XDRef overide defined

Dim labelMsg As String

labelMsg = "There is no label style XDRef override for point group " & _
ptGrp.GroupName & "."

'Test if XDRef label style is defined and if true define new message If ptGrp.LabelStyleXDRef ""Then

labelMsg = "The label style XDRef override for point group " & _
ptGrp.GroupName & " is: " & ptGrp.LabelStyleXDRef
End If

MsgBox labelMsg, vbInformation, "LabelStyleXDRef Example"

^{&#}x27; project database point group

LastUsedDwg Example

Sub Example_LastUsedDwg()

' This example returns the LastUsedDwg setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for LastUsedDrawing is: " & prefUser.LastUsedDrawing is: " & prefUser.Las

LastUsedDwgPath Example

Sub Example_LastUsedDwgPath()

' This example returns the LastUsedDwgPath setting. Dim prefUser As AeccPreferencesUser Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for LastUsedDwgPath is: " & prefUser.LastUsedD vbInformation, "LastUsedDwgPath Example"

LastUsedProj Example

Sub Example_LastUsedProj()

' This example returns the LastUsedProj setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for LastUsedProj is: " & prefUser.LastUsedProj, _ vbInformation, "LastUsedProj Example"

LastUsedProjPath Example

Sub Example_LastUsedProjPath()

' This example returns the LastUsedProjPath setting. Dim prefUser As AeccPreferencesUser Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for LastUsedProjPath is: " & prefUser.LastUsedPr vbInformation, "LastUsedProjPath Example"

Latitude Example

Sub Example_Latitude ()

- 'This example returns the Latitude and Longitude
- ' for the first CogoPoint in the collection.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Latitude for the first CogoPoint in the collection is: " & cogoPnt "The Longitude for the first CogoPoint in the collection is: " & cogoPnt.Lon

LayerFile Example

```
Sub Example_LayerFile()
```

```
'This example displays the LayerFile setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

Dim layerfile As String
layerfile = dbPref.LayerFile

'Test if layer file is set

If layerfile = "" Then
layerfile = "not set"

End If
```

MsgBox "The current value for LayerFile is: " & layerfile, vbInformation, "Lay

LayerStandard Example

```
Sub Example_LayerStandard()
```

```
'This example displays the LayerStandard setting for the current drawing. Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences

Dim laystandard As String
laystandard = dbPref.layerstandard

'Test if layer standard is set
If laystandard = "" Then
laystandard = "not set"
End If

MsgBox "The current value for LayerStandard is: " & laystandard, _
vbInformation, "LayerStandard Example"
```

LeftWidth Example

Sub Example_LeftWidth()

- ' This example returns the left width for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks

Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The left width for the alignment cross section at station " & station 8 Format(alignXSect.LeftWidth, "0.00"), vbInformation, "LeftWidth Example

Length Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 ParcelEntity, ParcelCurve, ParcelLine

Sub Example_Length_AlignEntity()

- 'This example returns the Length for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The Length of the first entity in the Alignment is: " & alignEnt.Leng vbInformation, "Length Example"

End Sub

Sub Example_Length_ParcelEntity()

- 'This example returns the Length for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The Length of the first entity in the Parcel is: " & parcelEnt.Length, vbInformation, "Length Example"

LExt Example

```
Sub Example_LExt()
```

```
'This example returns the Lext value for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The Lext value for the first Spiral in the alignment is: " & ali

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "LExt Example"
```

LinearDisplayFormat Example

Sub Example_LinearDisplayFormat()

```
'This example returns the LinearDispalyFormat setting for the current drawing Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences
```

```
'Convert the linear display format to a string.

Dim linearformat As String

Select Case dbPref.LinearDisplayFormat

Case 1
    linearformat = "Scientific"

Case 2
    linearformat = "Decimal"

Case 3
    linearformat = "Engineering"

Case 4
    linearformat = "Architectural"

Case 5
    linearformat = "Fractional"

End Select
```

MsgBox "The current value for LinearDisplayFormat is: " & linearformat, vbIı

LinearPrecision Example

Sub Example_LinearPrecision()

'This example returns the LinearPrecision setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for LinearPrecision is: " & dbPref.LinearPrecision

LinearUnit Example

```
Sub Example_LinearUnit()
```

```
'This example returns the LinearUnit setting for the current drawing.
Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences
'Convert the volume display unit to a string.
Dim unit As String
Select Case dbPref.LinearUnit
Case aecUnitInch
  unit = "inch"
Case aecUnitFoot
  unit = "foot"
Case aecUnitYards
  unit = "yard"
Case aecUnitMil
  unit = "millimeters"
Case aecUnitCentimeter
  unit = "centimeters"
Case aecUnitDecimeter
  unit = "decimeters"
Case aecUnitMeter
  unit = "meters"
End Select
```

MsgBox "The current value for LinearUnit is: " & unit, vbInformation, "Linear

LineIntersection Example

Sub Example_LineIntersection()

```
'This example gets the LineIntersection from a user supplied point
```

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim pnt1 As Variant Dim pnt2 As Variant

Dim coords As Variant

' Get the points

pnt1 = ThisDrawing.Utility.GetPoint(, "Enter the first point near the first align
pnt2 = ThisDrawing.Utility.GetPoint(, "Enter the second point near the first ali

'Get the line intersection location coords = align.LineIntersection(pnt1(0), pnt1(1), pnt2(0), pnt2(1))

MsgBox "The LineIntersection for the first Alignment is:" & vbCrLf & _

- " Station: " & coords(0) & vbCrLf & _
- " Direction: " & coords(1) & vbCrLf & _
- " Easting: " & coords(2) & vbCrLf & _
- " Northing: " & coords(3), vbInformation, "LineIntersection Example"

^{&#}x27; for the first Alignment in the collection..

LoadSetupProfile Example

Sub Example_LoadSetupProfile()

' This example loads an existing drawing setup profile.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.DatabasePreferences

' If no extension is supplied, ".set" is appended automatically. Pref.LoadSetupProfile("i20.set")

LockedPointNumbers Example

Sub Example_LockedPointNumbers()

```
'This examples locks a series of point numbers and
' then unlocks a subset of those point numbers.
Dim cogoPnts As AeccCogoPoints
Set cogoPnts = AeccApplication.ActiveProject.CogoPoints
'Create locked point string
Dim pntString As String
pntString = "1,4-8,40,10,12"
'Lock points
cogoPnts.LockPoints (pntString)
MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _
  vbInformation, "LockedPointNumbers Example"
' Create unlock point string
pntString = "4-8"
' Unlock points
cogoPnts.UnlockPoints (pntString)
MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _
```

vbInformation, "LockedPointNumbers Example"

LockPoints Example

```
Sub Example_LockPoints()
```

```
'This examples locks a series of point numbers and
' then unlocks a subset of those point numbers.
Dim cogoPnts As AeccCogoPoints
Set cogoPnts = AeccApplication.ActiveProject.CogoPoints
'Create locked point string
Dim pntString As String
pntString = "1,4-8,40,10,12"
'Lock points
cogoPnts.LockPoints (pntString)
MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _
  vbInformation, "LockPoints Example"
' Create unlock point string
pntString = "4-8"
' Unlock points
cogoPnts.UnlockPoints (pntString)
MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _
```

End Sub

vbInformation, "LockPoints Example"

LockType Example

```
Sub Example_LockType()
```

```
'This example returns the LockType setting for the
```

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

'Convert the lock type to a string.

Dim LockType As String

Select Case align.LockType

Case kNoLock

LockType = "no lock exists."

Case kReadLock

LockType = "the user can only read the object."

Case kWriteLock

LockType = "the user can read and write the object."

End Select

 $MsgBox \ "The \ current \ value \ for \ LockType \ is \ " \ \& \ LockType \ , \ vbInformation, "I$

^{&#}x27; first Alignment in the collection.

LOffset Example

```
Sub Example_LOffset()
```

```
'This example returns the LOffset for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The LOffset for the first Spiral in the alignment is: " & align.

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "LOffset Example"
```

Longitude Example

Sub Example_Longitude ()

- 'This example returns the Latitude and Longitude
- ' for the first CogoPoint in the collection.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Latitude for the first CogoPoint in the collection is: " & cogoPnt "The Longitude for the first CogoPoint in the collection is: " & cogoPnt.Lon

MaxElevation Example

Examples:

1 CrossSection

1 Surface

Sub Example_MaxElevation_CrossSection()

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The maximum elevation for the cross section at station " & station & Format(xSect.MaxElevation, "0.00"), vbInformation, "MaxElevation Examp

End Sub

Sub Example_MaxElevation_Surface()

'This example returns the MaxElevation for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MaxElevation for surface " & surf.Name & " is: " & Format(survbInformation, "MaxElevation Example"

^{&#}x27;This example returns the maximum elevation for the first cross section in the

^{&#}x27; cross section collection for the first alignment in the collection.

MaxFaceArea Example

Sub Example_MaxFaceArea()

' This example returns the MaxFaceArea for the first surface in the collection. Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MaxFaceArea for the first surface is: " & Format(surf.MaxFaceArea Example"

MagGrade Example

Sub Example_MaxGrade()

'This example returns the MaxGrade for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MaxGrade for the first surface is: " & Format(surf.MaxGrade, " vbInformation, "MaxGrade Example"

MaxOffset Example

Sub Example_MaxOffset()

'This example returns the maximum offset for the first cross section in the

' cross section collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The maximum offset for the cross section at station " & station & " is Format(xSect.MaxOffset, "0.00"), vbInformation, "MaxOffset Example"

MeanElelvation Example

Sub Example_MeanElevation()

' This example returns the MeanElevation for the first surface in the collection. Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MeanElevation for the first surface is: " & Format(surf.MeanElevbInformation, "MeanElevation Example"

MeasurementUnit Example

Sub Example_MeasurementUnit()

```
'This example returns the MeasurementUnit setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

'Convert the measurement unit to a string.

Dim unit As String

If dbPref.MeasurementUnit = acEnglish Then
    unit = "english"

Else
    unit = "metric"

End If

MsgBox "The current value for MeasurementUnit is " & unit, _
    vbInformation, "MeasurementUnit Example"
```

MidOrdinate Example

Sub Example_MidOrdinate()

```
'This example returns the MidOrdinate for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The MidOrdinate for the first Curve in the alignment is: " & Exit For End If

Next

MsgBox alignMsg, vbInformation, "MidOrdinate Example"
```

MinDepressionArea Example

Sub Example_MinDepressionArea()

'This example returns the MinDepressionArea setting for the WaterSheds.

Dim surf As AeccSurface

Dim wSheds As AeccWaterSheds

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wSheds = surf.Outputs.WaterSheds

MsgBox "The MinDepressionArea setting for WaterSheds is: " & wSheds.Min , vbInformation, "MinDepressionArea Example"

MinDepressionDepth Example

Sub Example_MinDepressionDepth()

'This example returns the MinDepressionDepth setting for the WaterSheds.

Dim surf As AeccSurface

Dim wSheds As AeccWaterSheds

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wSheds = surf.Outputs.WaterSheds

MsgBox "The MinDepressionDepth setting for WaterSheds is: " & wSheds.Mi , vbInformation, "MinDepressionDepth Example"

MinElevation Example

Examples:

1 CrossSection

1 Surface

Sub Example_MinElevation_CrossSection()

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The minimum elevation for the cross section at station " & station & Format(xSect.MinElevation, "0.00"), vbInformation, "MinElevation Exampl

End Sub

Sub Example_MinElevation_Surface()

'This example returns the MinElevation for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MinElevation for surface " & surf.Name & " is: " & Format(sur vbInformation, "MinElevation Example"

^{&#}x27;This example returns the minimum elevation for the first cross section in the

^{&#}x27; cross section collection for the first alignment in the collection.

MinFaceArea Example

Sub Example_MinFaceArea()

' This example returns the MinFaceArea for the first surface in the collection. Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MinFaceArea for the first surface is: " & Format(surf.MinFaceA vbInformation, "MinFaceArea Example"

MinGrade Example

Sub Example_MinGrade()

'This example returns the MinGrade for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The MinGrade for the first surface is " & Format(surf.MinGrade, "0. vbInformation, "MinGrade Example"

MinOffset Example

Sub Example_MinOffset()

'This example returns the minimum offset for the first cross section in the

' cross section collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The minimum offset for the cross section at station " & station & " is Format(xSect.MinOffset, "0.00"), vbInformation, "MinOffset Example"

Modified Example

class Module Code:

Public WithEvents cogoPnts As AeccCogoPoints

Private Sub cogoPnts_Modified()

MsgBox "Points have been modified", vbInformation, "Modified Example"

End Sub

Module Code:

Option Explicit
Dim eh As New EventHandler

Sub example_Modify_Event()

'This example establishes event handling for the CogoPoints object. Set eh.cogoPoints = AeccApplication.ActiveProject.CogoPoints

End Sub

Sub Example_Modify()

- 'This example change toggles the raw description of the first point
- ' in the collection to illustrate the Modify event.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication ActiveProject CogoP

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

'Get the points raw description
Dim strName As String
strName = cogoPnt.RawDescription

' Modify the raw description

```
If cogoPnt.RawDescription = "Old Description" Then
  cogoPnt.RawDescription = "New Description"
Else
  cogoPnt.RawDescription = "Old Description"
End If
```

' Show the override description on the command line ThisDrawing.Utility.Prompt vbCrLf & "The raw description of the first point is

Name Example

Examples:

- l AeccContourStyle
- 1 Alignment
- l CogoPoint
- 1 <u>CrossSectionBlock</u> (Civil Engineering Feature)
- 1 <u>CrossSectionSurface</u> (Civil Engineering Feature)
- 1 **DEMFile**
- 1 <u>DescriptionKeyFile</u>
- 1 Drawing
- l <u>FGProfiles</u> (Civil Engineering Feature)
- l Parcel
- l PointFile
- l PointGroup
- 1 PointGroupName
- 1 ProfileBlock (Civil Engineering Feature)
- l Project
- l Prototype
- l Surface

Sub Example_Name_AeccContourStyle()

'This example returns the Name for the contour style in the dictionary
Dim objContourStyle As AeccContourStyle
Set objContourStyle = ThisDrawing.Dictionaries("AECC_CONTOUR_STYL)

MsgBox "The value for the ContourStyle Name is: " & objContourStyle.Name

End Sub

Sub Example_Name_Alignment()

'This example returns the Name for the first Alignment in the collection Dim align As AeccAlignment Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The Name of the first Alignment is: " & align.Name, vbInformation,

End Sub

Sub Example_Name_CogoPoint()

'This example returns the Name for the first CogoPoint in the collection.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Name for the first CogoPoint in the collection is: " & cogoPnt.N

End Sub

Sub Example_Name_CrossSectionBlock()

- 'This example returns the name of the profile for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

 ${\bf Set\ align XSects} = {\bf Aecc Application. Active Document. Cross Section Blocks}$

Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The profile name for the alignment cross section at station " & station alignXSect.Name, vbInformation, "Name Example"

End Sub

Sub Example_Name_CrossSectionSurface()

- 'This example returns the name for the first cross
- ' section surface in the collection for the first cross section
- ' in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Dim xSectSurf As AeccCrossSectionSurface

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

Set xSectSurf = xSect.CrossSectionSurfaces.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The name for the first cross section surface for " & vbCrLf & _ "the cross section at station " & station & " is: " & _ xSectSurf.Name, vbInformation, "Name Example"

End Sub

Sub Example_Name_DEMFile()

- 'This example returns the Name for the first DEMFile in the
- ' first surface in the collection.

Dim surf As AeccSurface
Dim DEMFile As AeccDEMFile
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set DEMFile = surf.Inputs.DEMFiles.Item(0)

MsgBox "The Name for the first DEMFile is: " & DEMFile.Name, _ vbInformation, "Name Example"

End Sub

Sub Example_Name_DescriptionKeyFile()

'This function gets the DescriptionKeyFile name

Dim dKeyFile As AeccDescriptionKeyFile

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

MsgBox "The Name for the first DescriptionKeyFile in the collection is: " & c , vbInformation, "Name Example"

End Sub

Sub Example_Name_Drawing()

- 'This example returns the Name setting for the first Drawing
- ' in the collection

Dim dwg As AeccDrawing

Set dwg = AeccApplication.ActiveProject.Drawings(0)

MsgBox "The Name for the first Drawing in the collection is: " & dwg.Name _ , vbInformation, "Name Example"

End Sub

Sub Example_Name_FGProfiles()

- 'This example returns the Name setting of the FGProfile
- ' for the first alignment in the collection

Dim align As AeccAlignment
Dim FGProfs As AeccFGProfiles
Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Set FGProfs = align.FGProfiles

MsgBox "The Name of the finshed ground profile for the first Alignment is: " FGProfs.Name, vbInformation, "Name Example"

End Sub

Sub Example_Name_Parcel()

- 'This example starts by creating a Parcel named "NewParcel".
- 'The new Parcel name displayed. Finally, the new Parcel is renamed
- ' to "OldName" and is displayed again.

Dim parcels As AeccParcels

Dim parcel As AeccParcel

Set parcels = AeccApplication.ActiveProject.Parcels

' ' Add a new Parcel name "NewParcel" Set parcel = parcels.Add("NewParcel")

MsgBox "The Parcel name is: " & parcel.Name, vbInformation, "Name Examp

'' Rename the new Parcel to "OldParcel" parcels.Rename "NewParcel", "OldParcel"

MsgBox "The Parcel name is: " & parcel.Name, vbInformation, "Name Examp

End Sub

Sub Example_Name_PointFile()

- 'This example returns the Name setting for the first PointFile
- ' in the collection.

Dim surf As AeccSurface

Dim pntFile As AeccPointFile

```
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
Set pntFile = surf.Inputs.PointFiles.Item(0)
```

MsgBox "The Name for the first PointFile in the collection is: " & pntFile.Nan , vbInformation, "Name Example"

End Sub

Sub Example_Name_PointGroup()

- 'This function gets the Name for the first PointGroup
- ' in the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The Name for the first PointGroup is: " & _ pntGrp.Name, vbInformation, "Name Example"

End Sub

Sub Example_Name_PointGroupName()

- 'This example returns the Name setting for the first PointGroupName
- ' in the collection.

Dim surf As AeccSurface

Dim pntGrpName As AeccPointGroupName

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set pntGrpName = surf.Inputs.PointGroupNames.Item(0)

MsgBox "The Name for the first PointGroupName in the collection is: " & pnt, vbInformation, "Name Example"

End Sub

Sub Example_Name_ProfileBlock()

'This example returns the Name for the first ProfileBlock in the collection

Dim alignProf As AeccProfileBlock Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The Name of the first ProfileBlock in the collection is: " & alignProf vbInformation, "Name Example"

End Sub

Sub Example_Name_Project()

'This example returns the Name setting for the first Project

' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Name value for the first Project in the collection is: " & proj.Na

End Sub

Sub Example_Name_Prototype()

'This example returns the Name of the first prototype in the collection.

Dim prot As AeccPrototype

Set prot = AeccApplication.Prototypes.Item(0)

MsgBox "The Name of the first prototype in the collection is: " & prot.Name, "

End Sub

Sub Example_Name_Surface()

'This example returns the Name of the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The Name of the first surface is: " & surf.Name, vbInformation, "Na

NameOverride Example

Sub Example_NameOverride()

NameXDRef Example

Sub Example_NameXDRef()

'This function gets the Point group point name XDRef Dim ptGrp As AeccPointGroup Set ptGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The point group point NameXDRef is: " & ptGrp.NameXDRef, _ vbInformation, "NameXDRef Example"

NewProjectBased Example

Sub Example_NewProjectBased()

```
' This example creates a new project-based drawing.
Dim doc As AeccDocument
Dim newDoc As AeccDocument
Set doc = AeccApplication.ActiveDocument
```

' Specify the full path of the new drawing Dim strNewDwg As String strNewDwg = doc.Path + "\New Drawing"

' Specify the full path of the template
Dim strTemplate As String
strTemplate = AeccApplication.Application.Preferences.Files.TemplateDwgPa

Set newDoc = doc.NewProjectBased(strTemplate, strNewDwg)

MsgBox "The new drawing is named:" & newDoc.Name, vbInformation, "Nev

NextPointNumber Example

Sub Example_NextPointNumber()

' This gets the NextPointNumber in the CogoPoints collection Dim cogoPnts As AeccCogoPoints Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

MsgBox "The NextPointNumber for CogoPoints is: " & cogoPnts.NextPointNumber Example"

Normal Example

Sub Example_Normal()

```
'This example returns the normal value for the
```

' first face item in the collection.

Dim surf As AeccSurface

Dim face As AeccFace

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set face = surf.Outputs.Faces.Item(0)

Dim norm As Variant

norm = face.Normal

MsgBox "The Normal value for the first Face is " & norm(0) & ", " & norm(1) vbInformation, "Normal Example"

Northing Example

Examples:

- l AeccPoint
- l CogoPoint
- l TinPoint

Sub Example_Northing_AeccPoint()

- 'This example returns the Northing setting for the
- ' first Point object in a selection set

On Error Resume Next

- ' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete
- ' Create the selection set based on a point selection
- ' and filter for the Point objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AECC_POINT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode dataCode = dataValue ssetObj.SelectOnScreen groupCode, dataCode

Dim objPoint As AeccPoint Set objPoint = ssetObj.Item(0)

MsgBox "The setting for Northing is: " & objPoint.Northing, vbInformation, ".

End Sub

Sub Example_Northing_CogoPoint()

'This example returns the Northing for the first CogoPoint in the collection Dim cogoPnt As AeccCogoPoint Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Northing for the first CogoPoint in the collection is: " & cogoPn

End Sub

Sub Example_Northing_TinPoint()

- 'This example returns the Northing of the first
- 'TIN Point in the collection.

Dim surf As AeccSurface

Dim tin As AeccTinPoint

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set tin = surf.Outputs.TinPoints.Item(0)

MsgBox "The Northing of the first TIN Point is: " & tin.Northing, _ vbInformation, "Northing Example"

NorthRotation Example

Sub Example_NorthRotation()

'This example returns the NorthRotation setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for NorthRotation is: " & dbPref.NorthRotation, _ vbInformation, "NorthRotation Example"

Number Example

Examples:

- l AeccPoint
- 1 Alignment
- l CogoPoint
- 1 Parcel

Sub Example_Number_AeccPoint()

- 'This example returns the Number setting for the
- ' first Point object in a selection set

On Error Resume Next

- ' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete
- ' Create the selection set based on a point selection
- ' and filter for the Point objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AECC_POINT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode

```
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode
```

Dim objPoint As AeccPoint Set objPoint = ssetObj.Item(0)

MsgBox "The setting for Number is: " & objPoint.Number, vbInformation, "N

End Sub

Sub Example_Number_Alignment()

'This example returns the Number for the first Alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The Number of the first alignment is: " & align.Number, vbInformat

End Sub

Sub Example_Number_CogoPoint()

'This example returns the Number for the first CogoPoint in the collection.

Dim cogoPnt As AeccCogoPoint

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The Number for the first CogoPoint in the collection is: " & cogoPnt

End Sub

Sub Example_Number_Parcel()

'This example returns the parcel Number for the first parcel in the collection.

Dim parcel As AeccParcel

Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

MsgBox "The Number for the first Parcel in the collection is: " _

& parcel. Number, vbInformation, "Number Example"

NumberOfFaces Example

Sub Example_NumberOfFaces()

' This example returns the NumberOfFaces for the first surface in the collection Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The NumberOfFaces for the first surface is: " & surf.NumberOfFace vbInformation, "NumberOfFaces Example"

NumberOfPoints Example

Sub Example_NumberOfPoints()

' This example returns the NumberOfPoints for the first surface in the collectio Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The NumberOfPoints for the first surface is: " & surf.NumberOfPoir

ObjectID Example

```
Sub Example_ObjectID()
```

```
'This example returns the ObjectID value for the first Tangent found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Tangent entitiy in the first Alignment."

'Find first Tangent in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kTangent Then

alignMsg = "The ObjectID for the first Tangent in the alignment is: " & al

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "ObjectID Example"
```

Offset Example

Sub Example_Offset()

```
'This example returns the point code data for the first cross
' section surface in the collection for the first cross section
' in the collection.
Dim aligns As AeccAlignments
Dim align As AeccAlignment
Dim xSect As AeccCrossSection
Dim xSectPCode As AeccCrossSectionPointCode
Set aligns = AeccApplication.ActiveProject.Alignments
Set align = aligns. Item(0)
Set xSect = align.CrossSections.Item(0)
Set xSectPCode = xSect.CrossSectionPointCodes.Item(0)
' Get the alignment name
Dim alignName As String
alignName = align.Name
'Get the cross section station and format it
Dim station As String
station = aligns.DoubleToStaFormat(xSect.station)
MsgBox "The alignment name is: " & alignName & vbCrLf & _
  "The first cross section is at station: " & station & vbCrLf &
  "The data for the first point code is: " & vbCrLf &
  vbTab & "Code: " & xSectPCode.Code & vbCrLf & _
  vbTab & "Description: " & xSectPCode.Description & vbCrLf & _
  vbTab & "Elevation: " & Format(xSectPCode.elevation, "0.00") & vbCrLf &
  vbTab & "Offset: " & Format(xSectPCode.offset, "0.00"), _
  vbInformation, "Offset Example"
```

OffsetElevations Example

Sub Example_OffsetElevations()

```
'This example returns the first three offsets and elevations for the first cross
' section surface in the collection for the first cross section
' in the collection.
Dim aligns As AeccAlignments
Dim align As AeccAlignment
Dim xSect As AeccCrossSection
Dim xSectSurf As AeccCrossSectionSurface
Set aligns = AeccApplication.ActiveProject.Alignments
Set align = aligns.Item(0)
Set xSect = align.CrossSections.Item(0)
Set xSectSurf = xSect.CrossSectionSurfaces.Item(0)
' Get the alignment name
Dim alignName As String
alignName = align.Name
'Get the cross section station and format it
Dim station As String
station = aligns.DoubleToStaFormat(xSect.station)
'Get the cross section surface name
Dim surfName As String
surfName = xSectSurf.Name
' Get the offsets and elevations
Dim offElev As Variant
offElev = xSectSurf.OffsetElevations
MsgBox "The alignment name is: " & alignName & vbCrLf & _
  "The first cross section is at station: " & station & vbCrLf & _
  "The first surface name is: " & surfName & vbCrLf &
  "The first three offsets and elevations for the surface are:" & vbCrLf & _
```

```
vbTab & "Offset 1: " & Format(offElev(0), "0.00") & _ vbTab & "Elevation 1: " & Format(offElev(1), "0.00") & vbCrLf & _ vbTab & "Offset 2: " & Format(offElev(2), "0.00") & _ vbTab & "Elevation 2: " & Format(offElev(3), "0.00") & vbCrLf & _ vbTab & "Offset 3: " & Format(offElev(4), "0.00") & _ vbTab & "Elevation 3: " & Format(offElev(5), "0.00"), _ vbInformation, "OffsetElevation Example"
```

OffsetElevationToXy Example

Sub Example_OffsetElevationToXy()

```
'This example returns an Autocad XY for a given offset and elevation for the
```

Dim alignXSect As AeccCrossSectionBlock

Set alignXSect = AeccApplication.ActiveDocument.CrossSectionBlocks.Item(

Dim offElev(0 To 1) As Double

Dim XY As Variant

Dim station As String

Dim offset As Double

Dim elevation As Double

'Get the station for the first alignment cross section in the collection. station = alignXSect.station

'Get the offset and the elevation

offset = ThisDrawing.Utility.GetReal(vbCrLf & "Enter the offset in station " & elevation = ThisDrawing.Utility.GetReal("Enter the elevation in station " & sta

' Use the entered values to get the X and Y values

offElev(0) = offset

offElev(1) = elevation

XY = alignXSect.OffsetElevationToXy(offElev)

MsgBox "The X value for the offset is: " & Format(XY(0), "0.00") & vbCrLf {
"The Y value for the elevation is: " & Format(XY(1), "0.00") _
, vbInformation, "OffsetElevationToXy Example"

^{&#}x27; first alignment cross section in the collection.

Open Example

Sub Example_Open()

' This example opens an existing project-based drawing.
Dim dwgs As AeccDrawings
Set dwgs = AeccApplication.ActiveProject.Drawings

' Specify the full path of the drawing
Dim strDwg As String
Set strDwg = AeccApplication.ActiveProject.Drawings.Item(0).FullName

' Specify the full path of the project
Dim strProject As String
Set strProject = AeccApplication.ActiveProject.FullName

dwgs.Open strDwg, strProject

OpenProjectBased Example

Sub Example_OpenProjectBased()

```
'This example opens the first existing project-based drawing
```

Dim dwg As AeccDrawing

Dim doc As AeccDocument

Dim openDoc As AeccDocument

Set dwg = AeccApplication.ActiveProject.Drawings.Item(0)

Set doc = AeccApplication.ActiveDocument

' Specify the full path of the drawing

Dim strDwg As String

strDwg = dwg.FullName

Set openDoc = doc.OpenProjectBased(strDwg)

MsgBox "The name of the opened drawing is " & openDoc.Name, _ vbInformation, "OpenProjectBased Example"

^{&#}x27; in the collection.

Output Example

Sub Example_Output()

- 'This example returns the Count of Faces for the first Surface
- ' in the collection by using the Outputs property.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The number of Faces in the first surface is: " & surf.Outputs.Faces.C , vbInformation, "Outputs Example"

OverflowPoints Example

Sub Example_OverflowPoints()

- 'This example returns the first point in the OverflowPoints
- ' for the first WaterShed in the collection.

Dim surf As AeccSurface

Dim wShed As AeccWaterShed

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wShed = surf.Outputs.WaterSheds.Item(0)

Dim coords As Variant

coords = wShed.OverflowPoints

MsgBox "The first OverFlowPoint for the first WaterShed is: " & coords(0) & , vbInformation, "OverflowPoints Example"

OverrideDescription Example

Sub Example_OverrideDescription()

'This example returns the OverrideDescription for the first CogoPoint in the co

'This example the first point in the collection is in a Point Group Named "Exai

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

'Filter CogoPoints collection on GroupName cogoPnts.GroupName = "Example Group"

Set cogoPnt = cogoPnts.Item(0)

MsgBox "The OverrideDescription for the first CogoPoint in the collection is: cogoPnt.OverrideDescription, vbInformation, "OverrideDescription Exampl

OverrideElevation Example

Sub Example_OverrideElevation()

- 'This example returns the OverrideElevation for the first CogoPoint in the coll
- 'This example assumes the first point in the collection is in a
- ' Point Group named "Example Group".

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

'Filter CogoPoints collection on GroupName cogoPnts.GroupName = "Example Group"

Set cogoPnt = cogoPnts.Item(0)

MsgBox "The OverrideElevation for the first CogoPoint in the collection is: " cogoPnt.OverrideElevation, vbInformation, "OverrideElevation Example"

OverrideName Example

Sub Example_OverrideName()

- 'This example returns the OverrideName for the first CogoPoint in the collecti
- 'This example assumes the first point in the collection is in a
- ' Point Group named "Example Group".

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

'Filter CogoPoints collection on GroupName cogoPnts.GroupName = "Example Group"

Set cogoPnt = cogoPnts.Item(0)

MsgBox "The OverrideName for the first CogoPoint in the collection is: " & _ cogoPnt.OverrideName, vbInformation, "OverrideName Example"

OverrideNew Example

Sub Example_OverrideNew()

' This example returns the OverrideNew setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for OverrideNew is: " & prefUser.OverrideNew, _ vbInformation, "OverrideNew Example"

OverrideOpen Example

Sub Example_OverrideOpen()

' This example returns the OverrideOpen setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for OverrideOpen is: " & prefUser.OverrideOpen, vbInformation, "OverrideOpen Example"

Owner Example

Sub Example_Owner

'This example returns the Label for the first FileLock in the collection.

Dim filelock As AeccFileLock

Set filelock = AeccApplication.ActiveProject.fileLocks.Item(0)

MsgBox "The Label of the first FileLock in the collection is: " & filelock.Labe , vbInformation, "Label Example"

P Example

```
Sub Example_P()
```

```
'This example returns the P value for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The P value for the first Spiral in the alignment is: " & alignI

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "P Example"
```

Parcel Example

Sub Example_Parcel()

'This example returns the K Label layer for Parcel Preferences

' in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The K Label layer for Parcel Preferences in the current Project is: "_ & prefPrj.Parcel.GetString(kLabelLayer), vbInformation, "Parcel Example"

ParcelEntities Example

Sub Example_ParcelEntities()

'This example uses the ParcelEntities property to get the number of entities for

' of entities for the first parcel in the collection.

Dim parcel As AeccParcel

Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

MsgBox "The number of entites for the first Parcel in the collection is: "_ & parcel.ParcelEntities.Count, vbInformation, "ParcelEntities Example"

Paste Example

Sub Example_Paste()

```
'This example pastes the second surface in the collection to
```

' the first surface in the collection

Dim surf As AeccSurface

Dim surfs As AeccSurfaces

Set surfs = AeccApplication.ActiveProject.Surfaces

' Get the name of the second surface in the collection

Set surf = surfs.Item(1)

Dim surfName As String

surfName = surf.Name

' Set the current surface to the first surface in the collection

Set surf = surfs.Item(0)

surfs.CurrentSurface = surf.Name

' Paste the two surfaces surf.Paste surfName

MsgBox "The current surface name is: " & surfs.CurrentSurface & vbCrLf & _ "The pasted surface name is: " & surfName, vbInformation, "Paste Example

Path Example

Examples:

- 1 DescriptionKeyFile
- 1 DescriptionKeyFiles
- 1 Drawing
- 1 Drawings
- 1 FileLocks
- 1 Project

Sub Example_Path_DescriptionKeyFile()

- 'This example returns the Path for the first
- ' DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

MsgBox "The value for DescriptionKeyFile Path is: " & dKeyFile.Path, vbInfc

End Sub

Sub Example_Path_DescriptionKeyFiles()

'This example returns the Path for DescriptionKeyFiles.

Dim dKeyFiles As AeccDescriptionKeyFiles

Set dKeyFiles = AeccApplication.ActiveProject.DescriptionKeyFiles

MsgBox "The value for DescriptionKeyFiles Path is: " & dKeyFiles.Path, vbIn

Sub Example_Path_Drawing()

'This example returns the Path setting for the first Drawing

Dim dwg As AeccDrawing

Set dwg = AeccApplication.ActiveProject.Drawings(0)

MsgBox "The Path for the first Drawing in the collection is: " & dwg.Path _ , vbInformation, "Path Example"

End Sub

Sub Example_Path_Drawings()

'This example returns the Path for the drawings in the active project.

Dim dwgs As AeccDrawings

Set dwgs = AeccApplication.ActiveProject.Drawings

MsgBox "The Path value for Drawings is: " & dwgs.Path, vbInformation, "Patl

End Sub

Sub Example_Path_FileLocks()

'This example returns the Path for the FileLocks in the active project.

Dim fileLocks As AeccFileLocks

Set fileLocks = AeccApplication.ActiveProject.FileLocks

MsgBox "The Path value for FileLocks is: " & fileLocks.Path, vbInformation,

End Sub

Sub Example_Path_Project()

- 'This example returns the Path setting for the first Project
- ' in the collection

Dim proj As AeccProject

^{&#}x27; in the collection.

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The Path value for the first Project in the collection is: " & proj.Path,

Perimeter Example

Sub Example_Perimeter()

'This example returns the Perimeter for the first parcel in the collection. Dim parcel As AeccParcel Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

MsgBox "The Perimeter for the first Parcel in the collection is: "_ & Format(parcel.Perimeter, "0.00"), vbInformation, "Perimeter Example"

PerpIntersection Example

Sub Example_PerpIntersection()

```
'This example gets the PerpIntersection from a user supplied point
```

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim returnPnt As Variant Dim coords As Variant

' Get the point

returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point near the first alignment of the control of the contro

'Get the perpendicular intersection location coords = align.PerpIntersection(returnPnt(0), returnPnt(1))

MsgBox "The PerpIntersection for the first Alignment is:" & vbCrLf & _

- " Station: " & coords(0) & vbCrLf & _
- " Direction: " & coords(1) & vbCrLf & _
- " Easting: " & coords(2) & vbCrLf & _
- " Northing: " & coords(3), vbInformation, "PerpIntersection Example"

^{&#}x27; for the first Alignment in the collection..

PiEasting Example

```
Sub Example_PiEasting()
```

```
'This example returns the PiEasting for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The OiEasting for the first Curve in the alignment is: " & aliq Exit For
End If

Next

MsgBox alignMsg, vbInformation, "PiEasting Example"
```

PiNorthing Example

Sub Example_PiNorthing()

```
'This example returns the PiNorthing for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The PiNorthing for the first Curve in the alignment is: " & al
Exit For
End If

Next

MsgBox alignMsg, vbInformation, "PiNorthing Example"
```

PointByNumber Example

Sub Example_PointByNumber()

```
'This gets a CogoPoint given a point number.
Dim cogoPnts As AeccCogoPoints
Dim cogoPnt As AeccCogoPoint
Dim pntNum As Long
```

```
' Set point number to get to 10
pntNum = 10
Set cogoPnts = AeccApplication.ActiveProject.CogoPoints
Set cogoPnt = cogoPnts.PointByNumber(pntNum)
```

MsgBox "The Number for CogoPoints is: " & cogoPnt.Number, vbInformation

PointCodeDecription Example

Sub Example_PointCodeDescription()

```
'' This example returns the point code description for the first cross '' section in the collection.
```

Dim align As AeccAlignment
Dim xSects As AeccCrossSections
Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Set xSects = align.CrossSections

''Get the the description for point code 1
Dim desc As String
desc = xSects.PointCodeDescription(1)

MsgBox "The description for point code 1 is: " & _ "Description: " & desc, vbInformation, "PointCodeDescription Example"

PointFiles Example

Sub Example_PointFiles()

'This example returns the number of PointFiles in the first

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of PointFiles in the first Surface is: " & surfIn.PointFiles, vbInformation, "PointFiles Example"

^{&#}x27; surface in the collection.

PointGroupNames Example

Sub Example_PointGroupNames()

'This example returns the number of PointGroupNames in the first

Dim surf As AeccSurface

Dim surfIn As AeccSurfaceInputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfIn = surf.Inputs

MsgBox "The number of PointGroupNames in the first Surface is: " & surfIn.F , vbInformation, "PointGroupNames Example"

^{&#}x27; surface in the collection.

PointGroups Example

Sub Example_PointGroups()

- 'This example returns the number of PointGroups
- ' in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of PointGroups in the current Project is: " & proj.PointC , vbInformation, "PointGroups Example"

PointList Example

Sub Example_PointList()

'This function gets the PointList for the first PointGroup

' in the collection.

Dim pntGrp As AeccPointGroup

Set pntGrp = AeccApplication.ActiveProject.PointGroups.Item(0)

MsgBox "The PointList for the first PointGroup is: " & pntGrp.PointList, _ vbInformation, "PointList Example"

PointLocation Example

Sub Example_PointLocation()

```
'This example gets the PointLocation from user supplied values
```

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim station As Double

Dim offset As Double

Dim easting As Double

Dim northing As Double

Dim direction As Double

' Get the station and the offset

```
station = ThisDrawing.Utility.GetReal("Enter the station for the point: ")
offset = ThisDrawing.Utility.GetReal("Enter the offset for the point: ")
```

' Get the point location

align.PointLocation station, offset, easting, northing, direction

MsgBox "The Point Location is:" & vbCrLf & _

- " Easting: " & easting & vbCrLf & $_$
- " Northing: " & northing & vbCrLf & _
- " Direction: " & direction, vbInformation, "PointLocation Example"

^{&#}x27; for the first Alignment in the collection.

PointNameSize Example

Sub Example_PointNameSize()

' This gets the PointNameSize for the CogoPoints collection Dim cogoPnts As AeccCogoPoints Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

MsgBox "The PointNameSize for CogoPoints is: " & cogoPnts.PointNameSize vbInformation, "PointNameSize Example"

PointNumberSelect Example

Sub Example_PointNumberSelect()

dataCode = dataValue

ssetObj.SelectOnScreen groupCode, dataCode

```
'This gets all the CogoPoint numbers in a selection set and
' displays them as a comma delimited string.
Dim cogoPnts As AeccCogoPoints
Dim entity As Object
Dim pntNum As Long
Dim odjId As Long
Dim firstPnt As Boolean
Dim strPnts As String
Set cogoPnts = AeccApplication.ActiveProject.CogoPoints
On Error Resume Next
' Delete existing SelectionSet
ThisDrawing.SelectionSets("SSet").Delete
'Create the selection filtered for CogoPoint objects
Dim ssetObj As AcadSelectionSet
Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
Dim gpCode(0) As Integer
Dim dataValue(0) As Variant
gpCode(0) = 0
dataValue(0) = "AECC_POINT"
Dim groupCode As Variant
Dim dataCode As Variant
groupCode = gpCode
```

```
firstPnt = True
' Process selection set
For Each entity In ssetObj
  objId = entity.ObjectID
  pntNum = cogoPnts.PointNumberFromObjID(objId)
  If pntNum > 0 Then
    If firstPnt = True Then
       strPnts = Format(pntNum)
       firstPnt = False
    Else
       strPnts = strPnts + "," + Format(pntNum)
    End If
  End If
Next
MsgBox "The CogoPoint numbers in the selection are: " & _
  strPnts, vbInformation, "PointNumberFromObjID Example"
```

PointOnLineTolerance Example

Sub Example_PointOnLineTolerance()

' This example returns the PointOnLineTolerance for the first surface in the col Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The PointOnLineTolerance for the first surface is: " & surf.PointOnI vbInformation, "PointOnLineTolerance Example"

PointStringToArray Example

Sub Example_PointStringToArray()

PointTolerance Example

Sub Example_PointTolerance()

'This example returns the PointTolerance for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The PointTolerance for the first surface is: " & surf.PointTolerance, _ vbInformation, "PointTolerance Example"

Precision Example

Sub Example_Precision()

'This example returns the Precision for the first parcel in the collection. Dim parcel As AeccParcel Set parcel = AeccApplication.ActiveProject.parcels.Item(0)

MsgBox "The Precision for the first Parcel in the collection is: " _ & parcel.Precision, vbInformation, "Precison Example"

Preferences Example

Examples:

- 1 AeccApplication
- l Document
- 1 Project

Sub Example_Preferences_AeccApplication()

- 'This example returns the current setting of PrototypePath
- ' from the preferences object.

Dim preferences As AeccPreferences

Set preferences = AeccApplication.Preferences

MsgBox "The current value for PrototypePath is: " & preferences.Files.Prototy vbInformation, "Preferences Example"

End Sub

Sub Example_Preferences_Document()

'This example returns the AngularPrecision for the active document.

Dim doc As AeccDocument

Set doc = AeccApplication.ActiveDocument

MsgBox "The setting for AngularPrecision is: " & doc.Preferences.AngularPre vbInformation, "Preferences Example"

End Sub

Sub Example_Preferences_Project()

'This example returns the text size for Cogo Preferences

' in the current project.

Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The text size for Cogo Preferences in the current Project is: " & _ proj.Preferences.Cogo.GetDouble(kPntTextSize), vbInformation, "Preferences.Cogo.GetDouble(kPntTextSize), "Preferences.Cogo.GetDouble(kP

PreferencesPath Example

Sub Example_PreferencesPath()

' This example returns the PreferencesPath setting. Dim prefFiles As AeccPreferencesFiles Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for PreferencesPath is: " & prefFiles.PreferencesPath vbInformation, "PreferencesPath Example"

Profile Example

Sub Example_Profile()

'This example returns the base grid layer for Profile Preferences

' in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The base grid layer for Profle Preferences in the current Project is: " & prefPrj.Profile.GetString(kBaseGridLayer), vbInformation, "Profile Exam

ProfileBlocks Example

Sub Example_ProfileBlocks()

- 'This example returns the count of alignment profiles in
- ' the active document.

Dim doc As AeccDocument

Set doc = AeccApplication.ActiveDocument

MsgBox "The number of alignment profile blocks in the current document is: 'doc.ProfileBlocks.Count, vbInformation, "ProfileBlocks Example"

ProfileByType Example

Examples:

1 <u>EGProfiles</u> (Civil Engineering Feature)

l <u>FGProfiles</u> (Civil Engineering Feature)

Sub Example_ProfileByType_EGProfiles()

- 'This example gets the surface name of the first existing ground profile
- ' in the first alignment in the collection. An existing ground profile is set based
- ' upon the surface name with a center type.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Dim cenEGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

Dim surfName As String surfName = EGProf.SurfaceName

Set cenEGProf = align.EGProfiles.ProfileByType(kEgCenter, surfName)

MsgBox "The first station for the center existing ground profile is: "_ & cenEGProf.StationElevations(0), vbInformation, "ProfileByType Example

End Sub

Sub Example_ProfileByType_FGProfiles()

- 'This example uses the ProfileByType method to get the finished ground profil
- ' based on a center type, for the the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

```
Set FGProf = align.FGProfiles.Item(0)
```

Set FGProf = align.FGProfiles.ProfileByType(kEgCenter)

MsgBox "The Type of the finished ground profile is: "_ & FGProf.Type, vbInformation, "ProfileByType Example"

ProgramPath Example

Sub Example_ProgramPath()

' This example returns the ProgramPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for ProgramPath is: " & prefFiles.ProgramPath, _ vbInformation, "ProgramPath Example"

ProjectName Example

Sub Example_ProjectName()

'This example displays the ProjectName setting for the current drawing Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The setting for ProjectName is: " & dbPref.ProjectName, _ vbInformation, "ProjectName Example"

ProjectPath Example

Examples:

1 PreferencesFiles

1 Projects

Sub Example_ProjectPath_PreferencesFiles()

'This example returns the ProjectPath setting.

Dim prefFiles As AeccPreferencesFiles

Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for ProjectPath is: " & prefFiles.ProjectPath, _ vbInformation, "ProjectPath Example"

End Sub

Sub Example_ProjectPath_Projects()

'This example returns the ProjectPath setting.

Dim projs As AeccProjects

Set projs = AeccApplication.Projects

MsgBox "The ProjectPath value for Projects is: " & projs.ProjectPath, _ vbInformation, "ProjectPath Example"

Projects Example

Sub Example_Projects()

' This example returns the number of Projects at the ProjectPath.

Dim projs As AeccProjects

Set projs = AeccApplication.Projects

MsgBox "The number of projects at " & projs.ProjectPath & " is: " & projs.Co vbInformation, "Projects Example"

PrototypeName Example

Sub Example_PrototypeName()

'This example returns the PrototypeName setting for the first Project

' in the collection

Dim proj As AeccProject

Set proj = AeccApplication.Projects.Item(0)

MsgBox "The PrototypeName for the first Project in the collection is: " & proj , vbInformation, "PrototypeName Example"

PrototypePath Example

Sub Example_PrototypePath()

' This example returns the PrototypePath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for PrototypePath is: " & prefFiles.PrototypePath, vbInformation, "PrototypePath Example"

Prototypes Example

Sub Example_Prototypes()

' This example returns the number of Prototypes at the PrototypePath Dim prots As AeccPrototypes Set prots = AeccApplication.Prototypes

MsgBox "The number of prototypes is " & prots.Count

PVIs Example

Sub Example_PVIs()

- 'This example uses the PVIs property to get the number of PVIs for the
- ' finished ground profile in the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

MsgBox "The number of PVIs for the finished ground profile in the first alignr & FGProf.PVIs.Count, vbInformation, "PVIs Example"

RadialDistance Example

Sub Example_RadialDistance()

```
'This example returns the RadialDistance for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The RadialDistance for the first Spiral in the alignment is: "
Exit For
End If

Next

MsgBox alignMsg, vbInformation, "RadialDistance Example"
```

Radius Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_Radius_AlignCurve()
```

```
'This example returns the Radius for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The Radius for the first Curve in the alignment is: " _
& Format(alignEnt.Radius, "0.00")
Exit For
End If

Next
```

MsgBox alignMsg, vbInformation, "Radius Example"

End Sub

Sub Example_Radius_ParcelCurve()

- 'This example returns the Radius for the first curve in the
- ' first Parcel in the collection.

```
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The Radius for the first Curve in the Parcel is: " _
        & Format(parcelEnt.Radius, "0.00")
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "Radius Example"
```

RawDescription Example

Sub Example_RawDescription()

' This example returns the RawDescription for the first CogoPoint in the collec ${\sf Dim\ cogoPnt\ As\ AeccCogoPoint}$

Set cogoPnt = AeccApplication.ActiveProject.CogoPoints.Item(0)

MsgBox "The RawDescription for the first CogoPoint in the collection is: " & cogoPnt.RawDescription, vbInformation, "RawDescription Example"

RemoveAll Example

Examples:

- 1 Alignment
- 1 StationEquations

Sub Example_RemoveAll_Alignment()

'This example adds an Alignment made up of a tangent.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

' Add an Alignment named "Example Alignment" and starting at Station 50.0 Set align = aligns.Add("Example Alignment", 50#)

' Add a tangent

Dim tangent As AeccAlignTangent Set tangent = align.AddTangent(0#, 0#, 150#, 0#)

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti

'Remove the entities from the alignment align.RemoveAll

MsgBox "The total number of entities in the Alignment is: " & align.AlignEnti

End Sub

Sub Example_RemoveAll_StationEquations()

- 'This example removes all StationEquations from
- ' the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment Set aligns = AeccApplication.ActiveProject.Alignments

- 'Get the first alignment Set align = aligns.Item(0)
- ' Set the first alignment current. aligns.CurrentAlignment = align.Name
- ' Show then StationEquation count for the first alignment MsgBox "The StationEquation count for the first alignment is: " & align.Statio
- 'Remove all StationEquations align.StationEquations.RemoveAll
- ' Show the StationEquation count as zero after the remove all MsgBox "The StationEquation count for the first alignment is: " & align.Statio

RemoveAllLabels Example

Sub Example_RemoveAllLabels()

'This example removes all labels from a selected contour

```
On Error Resume Next
' Delete existing SelectionSet
ThisDrawing.SelectionSets("SSet").Delete
'Create the selection set based on a point selection
' and filter for Contour objects
Dim ssetObj As AcadSelectionSet
Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
Dim mode As Integer
Dim gpCode(0) As Integer
Dim dataValue(0) As Variant
gpCode(0) = 0
dataValue(0) = "AECC CONTOUR"
Dim groupCode As Variant
Dim dataCode As Variant
Dim returnPnt As Variant
groupCode = gpCode
dataCode = dataValue
returnPnt = ThisDrawing.Utility.GetPoint(, "Select a contour line: ")
ssetObj.SelectAtPoint returnPnt, groupCode, dataCode
```

Dim objContour As AeccContour Set objContour = ssetObj.Item(0)

obj Contour. Remove All Labels

RemoveLabelAt Example

Sub Example_RemoveLabelAt()

Set Ent = ssetObj.Item(0)

Ent.RemoveLabelAt returnPnt(0), returnPnt(1)

```
'This example removes a label from a Contour object by selecting the
' contour
On Error Resume Next
' Delete existing SelectionSet
ThisDrawing.SelectionSets("SSet").Delete
'Create the selection set based on a point selection
' and filter for Contour objects
Dim ssetObj As AcadSelectionSet
Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")
Dim mode As Integer
Dim gpCode(0) As Integer
Dim dataValue(0) As Variant
gpCode(0) = 0
dataValue(0) = "AECC_CONTOUR"
Dim groupCode As Variant
Dim dataCode As Variant
Dim returnPnt As Variant
groupCode = gpCode
dataCode = dataValue
returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point on a contour line: ")
ssetObj.SelectAtPoint returnPnt, groupCode, dataCode
Dim Ent As AeccContour
```

Rename Example

Examples:

1 Parcels

1 Surfaces

Sub Example_Rename_Parcels()

- 'This example starts by creating a Parcel named "NewParcel".
- 'The new Parcel name displayed. Finally, the new Parcel is renamed
- ' to "OldName" and is displayed again.

Dim parcels As AeccParcels

Dim parcel As AeccParcel

Set parcels = AeccApplication.ActiveProject.Parcels

'' Add a new Parcel name "NewParcel"
Set parcel = parcels.Add("NewParcel")

MsgBox "The Parcel name is: " & parcel.Name, vbInformation, "Rename Exar

'' Rename the new Parcel to "OldParcel" parcels.Rename "NewParcel", "OldParcel"

MsgBox "The Parcel name is: " & parcel.Name, vbInformation, "Rename Exar

End Sub

Sub Example_Rename_Surfaces()

'This example renames a the first Surface in the collection to "FinalSurface".

Dim surfs As AeccSurfaces

Dim surf As AeccSurface

Set surfs = AeccApplication.ActiveProject.Surfaces

'Get the name of the first surface in the collection Set surf = surfs.Item(0) Dim surfName As String surfName = surf.Name

'Rename the surface surfs.Rename surfName, "FinalSurface"

MsgBox "The Surface name FinalSurface was named: " & surfName, vbInforn

RevisionNumber Example

Sub Example_RevisionNumber()

'This example returns the RevisionNumber for the first surface in the collectio Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The RevisionNumber for the first surface is: " & surf.RevisionNumb

RightWidth Example

Sub Example_RightWidth()

- 'This example returns the right width for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks
Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The right width for the alignment cross section at station " & station Format(alignXSect.RightWidth, "0.00"), vbInformation, "RightWidth Exam

RotateByDescriptionParam Example

Sub Example_RotateByDescriptionParam()

- 'This example returns the RotateByDescriptionParam for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey RotateByDescriptionParam is: " & dK , vbInformation, "RotoateByDescriptionParam Example"

RotateByFixedFactor Example

Sub Example_RotateByFixedFactor()

- 'This example returns the RotateByFixedFactor for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey RotateByFixedFactor is: " & dKey.Ro vbInformation, "RotateByFixedFactor Example"

RotateClockwise Example

Sub Example_RotateClockwise()

- 'This example returns the RotateClockwise for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey RotateClockwise is: " & dKey.Rotate(, vbInformation, "RotateClockwise Example"

RotateDescriptionParam Example

Sub Example_RotateDescriptionParam()

- 'This example returns the RotateDescriptionParam for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey RotateDescriptionParam is: " & dKey. vbInformation, "RotateDescriptionParam Example"

RotateFixedFactor Example

Sub Example_RotateFixedFactor()

- 'This example returns the RotateFixedFactor for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey RotateFixedFactor is: " & dKey.Rotate, vbInformation, "RotateFixedFactor Example"

SampleElevation Example

Sub Example_SampleElevation()

```
'This example returns an array of points between a selected start and end point
```

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Dim pnt1 As Variant

Dim pnt2 As Variant

Dim derivedPnts As Variant

'Return points using a prompt

pnt1 = ThisDrawing.Utility.GetPoint(, "Select first point: ")

pnt2 = ThisDrawing.Utility.GetPoint(pnt1, "Select second point: ")

derivedPnts = surf.SampleElevations(pnt1, pnt2)

MsgBox "The first two points in the derived points are:" & vbCrLf & _ derivedPnts(0) & ", " & derivedPnts(1) & ", " & derivedPnts(2) & vbCrLf & derivedPnts(3) & ", " & derivedPnts(4) & ", " & derivedPnts(5), vbInformati "SampleElevations Example"

^{&#}x27; for the first surface in the collection. The array from points includes the start

^{&#}x27; and end points and all points that intersect an edge between the start and

^{&#}x27;endpoint.

Save Example

Examples:

- 1 Alignment
- l CogoPoint
- l DescriptionKey
- l PointGroup

Sub Example_Save_Alignment()

- 'This example changes the Description and StartingStation
- ' for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = Aligns.Item(0)

- ' Make the first alignment in the collection current aligns.CurrentAlignment = align.Name
- ' Set AutoSave to FALSE to prevent writing the changes
- ' until they are all made

aligns.AutoSave = False

- ' Make changes align.Description = "New Description" align.StartingStation = 20#
- ' Save changes align.Save
- 'Reset AutoSave to the default state of TRUE aligns.AutoSave = True

MsgBox "The first Alignments Description is: " & align.Description & vbCrLf "The first Alignments StartingStation is: " & align.StartingStation, _ vbInformation, "Save Example"

End Sub

```
Sub Example_Save_CogoPoint()
```

```
'This example changes the Northing, Easting, and Elevation
```

' for the first CogoPoint in the collection.

Dim cogoPnts As AeccCogoPoints

Dim cogoPnt As AeccCogoPoint

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

Set cogoPnt = cogoPnts.Item(0)

'Set AutoSave to FALSE to prevent writing the changes

' until they are all made

cogoPnts.AutoSave = False

' Make changes

cogoPnt.Northing = 100#

cogoPnt.Easting = 100#

cogoPnt.Elevation = 150#

' Save changes

cogoPnt.Save

'Reset AutoSave to the default state of TRUE cogoPnts.AutoSave = True

MsgBox "The following changes were made to CogoPoint " & cogoPnt.Numb

- " Northing: " & cogoPnt.Northing & vbCrLf & _
- " Easting: " & cogoPnt.Easting & vbCrLf & _
- " Elevation: " & cogoPnt.Elevation, vbInformation, "Save Example"

Sub Example_Save_DescriptionKey()

```
'This example changes the layer settings
```

Dim dKeyFiles As AeccDescriptionKeyFiles

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFiles = AeccApplication.ActiveProject.DescriptionKeyFiles

Set dKeyFile = dKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

' until they are all made

dKeyFiles.AutoSave = False

' Make changes

dKey.DescriptionLayer = "dKey Desc"

dKey.SymbolLayer = "dKey Symb"

' Save changes

dKey.Save

' Reset AutoSave to the default state of TRUE

dKeyFiles.AutoSave = True

MsgBox "The DescriptionLayer is: " & dkey.DescriptionLayer & vbCrLf & _ "The SymbolLayer is: " & dkey.SymbolLayer, vbInformation, "Save Examp

End Sub

Sub Example_Save_PointGroup()

- 'This example changes the Description and Elevation
- ' for the first PointGroup in the collection.

Dim pntGrps As AeccPointGroups

Dim pntGrp As AeccPointGroup

^{&#}x27; for the first DescriptionKey in the collection.

^{&#}x27; Set AutoSave to FALSE to prevent writing the changes

Set pntGrps = AeccApplication.ActiveProject.PointGroups
Set pntGrp = pntGrps.Item(0)

- ' Set AutoSave to FALSE to prevent writing the changes
- ' until they are all made pntGrps.AutoSave = False
- ' Make changes pntGrp.Description = "New Description" pntGrp.Elevation = 100#
- ' Save changes pntGrp.Save
- 'Reset AutoSave to the default state of TRUE pntGrps.AutoSave = True

MsgBox "The first PointGroup Description is: " & pntGrp.Description & vbCr "The first PointGroup Elevation is: " & pntGrp.Elevation, vbInformation, "S

SaveAsDefault Example

Sub Example_SaveAsDefault()

```
'This example sets the TextHeight setting to 0.25
```

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

dbPref.TextHeight = 0.25 dbPref.SaveAsDefault

MsgBox "The new default value for TextHeight is: " & dbPref.TextHeight, _ vbInformation, "SaveAsDefault Example"

^{&#}x27; and saves it as the default.

SaveSetupProfile Example

Sub Example_SaveSetupProfile()

' This example saves a new drawing setup profile.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.DatabasePreferences

'The file extension ".set" is appended automatically. dbPref.SaveSetupProfile("myprofile")

SectionVolume Example

Sub Example_SectionVolume()

```
'This example calculates the volume data for the second cross section in the
' cross section collection for the first alignment in the collection.
Dim aligns As AeccAlignments
Dim align As AeccAlignment
Dim xSect As AeccCrossSection
Set aligns = AeccApplication.ActiveProject.Alignments
Set align = aligns.Item(0)
Set xSect = align.CrossSections.Item(0)
'Get the cross section station and format it
Dim station As String
station = aligns.DoubleToStaFormat(xSect.station)
' Define result variables
Dim cutArea As Double
Dim cutCentroid As Double
Dim cutVolume As Double
Dim fillArea As Double
Dim fillCentroid As Double
Dim fillVolume As Double
' Get volume data
xSect.SectionVolume kVolumeByAvgEndArea, 1#, 1#, 1#, cutArea, cutCentroi
MsgBox "The volume data at station " & station & " is: " & vbCrLf & _
  "Cut Area: " & Format(cutArea, "0.00") & vbCrLf &
  "Cut Centroid: " & Format(cutCentroid, "0.00") & vbCrLf & _
  "Cut Volume: " & Format(cutVolume, "0.00") & vbCrLf &
  "Fill Area: " & Format(fillArea, "0.00") & vbCrLf &
  "Fill Centroid: " & Format(fillCentroid, "0.00") & vbCrLf & _
```

"Fill Volume: " & Format(fillVolume, "0.00"), vbInformation, "SectionVolur

ScaleByDescriptionParam Example

Sub Example_ScaleByDescriptionParam()

- 'This example returns the ScaleByDescriptionParam for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleByDescriptionParam is: " & dK ϵ , vbInformation, "ScaleByDescriptionKey Example"

ScaleByDrawingScale Example

Sub Example_ScaleByDrawingScale()

- 'This example returns the ScaleByDrawingScale for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleByDrawingScale is: " & dKey.So vbInformation, "ScaleByDrawingScale Example"

ScaleByFixedFactor Example

Sub Example_ScaleByFixedFactor()

- 'This example returns the ScaleByFixedFactor for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleByFixedFactor is: " & dKey.ScaleByFixedFactor Example"

ScaleDescriptionParam Example

Sub Example_ScaleDescriptionParam()

- 'This example returns the ScaleDescriptionParam for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleDescriptionParam is: " & dKey.S , vbInformation, "ScaleDescriptionParam Example"

ScaleFixedFactor Example

Sub Example_ScaleFixedFactor()

- 'This example returns the ScaleFixedFactor for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleFixedFactor is: " & dKey.ScaleF , vbInformation, "ScaleFixedFactor Example"

ScaleInXY Example

Sub Example_ScaleInXY()

- 'This example returns the ScaleInXY for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleInXY is: " & dKey.ScaleInXY, _ vbInformation, "ScaleInXY Example"

ScaleInZ Example

Sub Example_ScaleInZ()

- 'This example returns the ScaleInZ for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey ScaleInZ is: " & dKey.ScaleInZ, _ vbInformation, "ScaleInZ Example"

ScaleOnInsert Example

Sub Example_ScaleOnInsert()

```
'This example returns the ScaleOnInsert setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

'Convert the scale on insert value to a string.

Dim scaleins As String

If dbPref.ScaleOnInsert = True Then
    scaleins = "scale on insert."

Else
    scaleins = "do not scale on insert."

End If

MsgBox "The current value for ScaleOnInsert is " & scaleins, _
    vbInformation, "ScaleOnInsert Example"
```

SearchType Example

```
Sub Example_SearchType()
```

```
'This example returns how the Faces collection was generated.

Dim surf As AeccSurface

Dim faces As AeccFaces

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set faces = Surf.Outputs.Faces
```

Dim msgStr As String

```
Select Case faces.SearchType
Case kNoSearch
  msgStr = "No search"
Case kSearchByBoundar
  msgStr = "Search by boundary"
Case kSearchByPath
  msgStr = "Search by path"
Case kSearchByPoint
  msgStr = "Search by point"
End Select
```

MsgBox "The Faces search type is set to: " & msgStr, vbInformation, "Search1

SectionByStation Example

Sub Example_SectionByStation()

'This example returns a cross section by providing the station

' to the first alignment in the collection.

Dim align As AeccAlignment

Dim xSects As AeccCrossSections

Dim xSect As AeccCrossSection

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set xSects = align.CrossSections

'Get the station for the first cross section in the cross section collection Dim station As Double station = xSects.Item(0).station

' Get the cross section block for the station set above Set xSect = xSects.SectionByStation(station)

MsgBox "The maimum elevation for the cross section at station " & station & Format(xSect.MaxElevation, "0.00"), vbInformation, "SectionByStation Exa

SetBoundingBox Example

Sub Example_SetBoundingBox()

```
'This example sets a surface bounding box for the first surface
'in the collection, based upon two picked points.

Dim surf As AeccSurface
Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Dim pnt1 As Variant
Dim pnt2 As Variant

pnt1 = ThisDrawing.Utility.GetPoint(, "Select first corner: ")
pnt2 = ThisDrawing.Utility.GetCorner(Pnt1, "Select other corner: ")

surf.SetBoundingBox pnt1, pnt2
```

SetDouble Example

Sub Example_SetDouble()

- 'This example uses SetDouble to define the ContLabelSpacingDist
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

surfPref.SetDouble kContLabelSpacingDist, 200#

MsgBox "The value for ContLabelSpacingDist is " & surfPref.GetDouble(kCo vbInformation, "SetDouble Example"

SetInteger Example

Sub Example_SetInteger()

- 'This example uses SetInteger to define the ContLabelPrecision
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

surfPref.SetInteger kContLabelPrecision, 3

MsgBox "The value for ContLabelPrecision is " & surfPref.GetInteger(kContL vbInformation, "SetInteger Example"

SetReferenceCurve Example

Sub Example_SetReferenceCurve()

```
'This example uses the SetReferenceCurve method to attach curve text
' to a selected arc or circle
Dim ent As AcadEntity
Dim basePnt As Variant
' Pick the curve
On Error Resume Next
RETRY:
Err.Clear
ThisDrawing.Utility.GetEntity ent, basePnt, "Select a curve"
If Err <> 0 Then
  ThisDrawing.Utility.Prompt "No entity found" + vbCrLf
  GoTo RETRY
End If
If ent.ObjectName <> "AcDbCircle" And _
  ent.ObjectName <> "AcDbArc" Then
  ThisDrawing.Utility.Prompt "Entity is not an arc or circle" + vbCrLf
  GoTo RETRY
End If
' Attach the curve text
Dim ctext As AeccCurveText
Set ctext = ThisDrawing.ModelSpace.AddCustomObject("AecDbCurveText")
ctext.SetReferenceCurve ent
ctext.TextAbove = "ABOVE"
ctext.TextBelow = "BELOW"
```

SetString Example

Sub Example_SetSting()

- 'This example uses SetString to define the SurfaceLayer
- ' for PreferencesSurface.

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

surfPref.SetString kSurfaceLayer, "MyLayer"

MsgBox "The value for Surface Layer is " & surfPref. GetString (kSurface Layer layer) where the surface of th

SheetHeight Example

Sub Example_SheetHeight()

'This example returns the SheetHeight setting for the current drawing. Dim dbPref As AeccDatabasePreferences Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for SheetHeight is: " & dbPref.SheetHeight, _ vbInformation, "SheetHeight Example"

SheetWidth Example

Sub Example_SheetWidth()

' This example returns the SheetWidth setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current value for SheetWidth is " & dbPref.SheetWidth, _ vbInformation, "SheetWidth Example"

ShortTangent Example

Sub Example_ShortTangent()

```
'This example returns the ShortTangent for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The ShortTangent for the first Spiral in the alignment is: " & Exit For End If

Next

MsgBox alignMsg, vbInformation, "ShortTangent Example"
```

ShowStartUpDialog Example

Sub Example_ShowStartupDialog()

' This example returns the ShowStartupDialog setting.
Dim prefUser As AeccPreferencesUser
Set prefUser = AeccApplication.Preferences.User

MsgBox "The current value for ShowStartupDialog is: " & prefUser.ShowStart

ShowSubFolders Example

Sub Example_ShowSubfolders()

'This example returns the ShowSubfolders setting for the active project. Dim dwgs As AeccDrawings

Set dwgs = AeccApplication.ActiveProject.Drawings

MsgBox "The ShowSubfolders value for Drawings in the current Project is " & , vbInformation, "ShowSubfolders Example"

SpeedtablesPath Example

Sub Example_SpeedTablesPath()

' This example returns the SpeedTablesPath setting. Dim prefFiles As AeccPreferencesFiles Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for SpeedTablesPath is: " & prefFiles.SpeedTables vbInformation, "SpeedTablesPath Example"

SpiEasting Example

```
Sub Example_SpiEasting()
```

```
'This example returns the SpiEasting for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The SpiEasting for the first Spiral in the alignment is: " & aliexit For
End If

Next

MsgBox alignMsg, vbInformation, "SpiEasting Example"
```

SpilTangent Example

```
Sub Example_SpilTangent()
```

```
'This example returns the SpilTangent for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment
Dim alignEnts As AeccAlignEntity
Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment
For Each alignEnt In align.AlignEntities
If alignEnt.Type = kSpiral Then
alignMsg = "The SpilTangent for the first Spiral in the alignment is: " & a
Exit For
End If
Next

MsgBox alignMsg, vbInformation, "SpilTangent Example"
```

SpiNorthing Example

Sub Example_SpiNorthing()

```
'This example returns the SpiNorthing for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then
alignMsg = "The SpiNorthing for the first Spiral in the alignment is: " & a
Exit For
End If

Next

MsgBox alignMsg, vbInformation, "SpiNorthing Example"
```

SpiralType1 Example

```
Sub Example_SpiralType1()
```

```
'This example returns the SpiralType1 for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The SpiralType1 for the first Spiral in the alignment is: " & a

Exit For

End If

Next

MsgBox alignMsg, vbInformation, "SpiralType1 Example"
```

SpiralType2 Example

```
Sub Example_SpiralType2()
```

```
'This example returns the SpiralType2 for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
        alignMsg = "The SpiralType2 for the first Spiral in the alignment is: " & a
        Exit For
    End If

Next

MsgBox alignMsg, vbInformation, "SpiralType2 Example"
```

StartDirection Example

Examples:

- 1 AlignCurve
- 1 AlignSpiral

```
Sub Example_StartDirection_AlignCurve()
```

```
'This example returns the StartDirection for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment
```

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication ActiveProject Ali

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

```
Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."
```

```
' Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then

alignMsg = "The StartDirection for the first Curve in the alignment is: " 8

Exit For

End If
```

MsgBox alignMsg, vbInformation, "StartDirection Example"

End Sub

Next

Sub Example_StartDirection_AlignSpiral()

- 'This example returns the StartDirection for the first Spiral found in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

```
Dim alignEnts As AeccAlignEntity
Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

' Find first Spiral in the alignment
For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
        alignMsg = "The StartDirection for the first Spiral in the alignment is: " &
        Exit For
    End If
Next

MsgBox alignMsg, vbInformation, "StartDirection Example"
```

StartEasting Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 ParcelEntity, ParcelCurve, ParcelLine

Sub Example_StartEasting_AlignEntity()

- 'This example returns the StartEasting for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The StartEasting of the first entity in the Alignment is: " & alignEnt. vbInformation, "StartEasting Example"

End Sub

Sub Example_StartEasting_ParcelEntity()

- 'This example returns the StartEasting for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The StartEasting of the first entity in the Parcel is: " & parcelEnt.StavbInformation, "StartEasting Example"

StartingStation Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 Alignment
- 1 ProfileBlock (Civil Engineering Feature)

Sub Example_StartingStation_AlignEntity()

- 'This example returns the StartingStation for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The StartingStation of the first entity in the Alignment is: " & _ alignEnt.StartingStation, vbInformation, "StartingStation Example"

End Sub

Sub Example_StartingStation_Alignment()

- 'This example returns the StartingStation for the first alignment
- ' in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The StartingStation of the first Alignment is: " & align.StartingStatic vbInformation, "StartingStation Example"

Sub Example_StartingStation_ProfileBlock()

'This example returns the StartingStation for the first ProfileBlock

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The StartingStation for the first ProfileBlock in the collection is: " _ & alignProf.StartingStation, vbInformation, "StartingStation Example"

^{&#}x27; in the collection

StartNorthing Example

Examples:

- 1 AlignEntity, AlignCurve, AlignSpiral, AlignTangent
- 1 ParcelEntity, ParcelCurve, ParcelLine

Sub Example_StartNorthing_AlignEntity()

- 'This example returns the StartNorthing for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

MsgBox "The StartNorthing of the first entity in the Alignment is: " & alignEn vbInformation, "StartNorthing Example"

End Sub

Sub Example_StartNorthing_ParcelEntity()

- 'This example returns the StartNorthing for the first entity in the
- ' first Parcel in the collection.

Dim parcel As AeccParcel

Dim parcelEnt As AeccParcelEntity

Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)

Set parcelEnt = parcel.ParcelEntities.Item(0)

MsgBox "The StartNorthing of the first entity in the Parcel is: " & parcelEnt.St vbInformation, "StartNorthing Example"

Station Example

Examples:

- 1 <u>CrossSection</u> (Civil Engineering Feature)
- 1 <u>CrossSectionBlock</u> (Civil Engineering Feature)
- 1 PVI (Civil Engineering Feature)
- 1 <u>Superelevation</u> (Civil Engineering Feature)

Sub Example_Station_CrossSection()

- 'This example returns the station for the first cross section in the
- ' cross section collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = aligns.Item(0)

Set xSect = align.CrossSections.Item(0)

'Get the cross section station and format it

Dim station As String

station = aligns.DoubleToStaFormat(xSect.station)

MsgBox "The station for the first cross section in the collection is: " & _ station, vbInformation, "Station Example"

End Sub

Sub Example_Station_CrossSectionBlock()

- 'This example returns the station for the
- ' first alignment cross section in the collection.

Dim aligns As AeccAlignments

Dim alignXSects As AeccCrossSectionBlocks
Dim alignXSect As AeccCrossSectionBlock
Set aligns = AeccApplication.ActiveProject.Alignments
Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks
Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = aligns.DoubleToStaFormat(alignXSect.station)

MsgBox "The station for the first alignment cross section in the collection is: " station, vbInformation, "Station Example"

End Sub

Sub Example_Station_PVI()

- 'This example returns the Station value for the second PVI in the
- ' first finished ground of the first alignment in the collection.

Dim align As AeccAlignment

Dim FGProf As AeccFGProfile

Dim PVI As AeccPVI

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set FGProf = align.FGProfiles.Item(0)

Set PVI = FGProf.PVIs.Item(1)

MsgBox "The Station of the second PVI is: "_ & Format(PVI.Station, "0.000"), vbInformation, "Station Example"

End Sub

Sub Example_Station_Superelevation()

- 'This example returns the data for the first superelevation in the
- ' superelevation collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim sElev As AeccSuperelevation
Set aligns = AeccApplication.ActiveProject.Alignments
Set align = AeccApplication.ActiveProject.Alignments.Item(0)
Set sElev = align.Superelevations.Item(0)

' Get the alignment name Dim alignName As String alignName = align.Name

'Get the superelevation station and format it Dim station As String station = aligns.DoubleToStaFormat(sElev.station)

' Get the superelevation curve code
Dim crvCode As String
Select Case sElev.curveCode
Case kSERightHandCurve
crvCode = "Right Hand Curve"
Case kSELeftHandCurve
crvCode = "Left Hand Curve"
End Select

'Get the superelevation code

Dim supCode As String

Select Case sElev.SuperelevationCode

Case kSEFullCrown

supCode = "Full Crown"

Case kSEHalfCrown

supCode = "Half Crown"

Case kSECrownRemoved

supCode = "Crown Removed"

Case kSEFullSuperelevations

supCode = "Full Superelevation"

Case kSEReverseCurve

supCode = "Reverse Curve"

Case kSECompoundCurve

supCode = "Compound Curve"

End Select

```
MsgBox "The alignment name is: " & alignName & vbCrLf & _
"The data for the first superelevation is: " & vbCrLf & _
vbTab & "Station: " & station & vbCrLf & _
vbTab & "Curve Code: " & crvCode & vbCrLf & _
vbTab & "Superelevation Code: " & supCode, _
vbInformation, "Station Example"
```

StationAhead Example

Sub Example_StationAhead()

- 'This example returns the StationAhead value
- ' for the StationEquation in the collection.

Dim align As AeccAlignment

Dim StaEqu As AeccStationEquation

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set StaEqu = align.StationEquations.Item(0)

MsgBox "The setting StationAhead is " & StaEqu.StationAhead, vbInformatio

StationBack Example

Sub Example_StationBack()

- 'This example returns the StationBack value
- ' for the StationEquation in the collection.

Dim align As AeccAlignment

Dim StaEqu As AeccStationEquation

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set StaEqu = align.StationEquations.Item(0)

MsgBox "The setting StationBack is " & StaEqu.StationBack, vbInformation, '

StationElevations Example

Sub Example_StationElevations()

```
'This example uses StationElevations to displays the station and elevation
```

' for the first existing ground profile in the first alignment in the collection.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

```
MsgBox "The first station for the first existing ground profile is: " _ & Format(EGProf.StationElevations(0), "0.00") & vbCrLf & _ "The first elevation for the first existing ground profile is: " _ & Format(EGProf.StationElevations(1), "0.00") & vbCrLf, _ vbInformation, "StaionElevations Example"
```

StationElevationToXy Example

Sub Example_StationElevationToXy()

```
'This example returns the StationElevationToXy for the first ProfileBlock
' in the collection.
Dim alignProf As AeccProfileBlock
Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)
Dim StaElev(0 To 1) As Double
Dim XY As Variant
Dim station As Double
Dim elevation As Double
'Get the station and the elevation
station = ThisDrawing.Utility.GetReal("Enter the station for the X value: ")
elevation = ThisDrawing.Utility.GetReal("Enter the elevation for the Y value: '
'Use the entered values to get the X and Y values
staElev(0) = station
staElev(1) = elevation
XY = alignProf.StationElevationToXy(staElev)
MsgBox "The X value for the StationElevation is: " & Format(XY(0), "0.00")
  "The Y value for the StationElevation is: " & Format(XY(1), "0.00")
  , vbInformation, "StationElevationToXy Example"
```

StationExample Example

Sub Example_StationEquations()

- 'This example returns the number of StationEquations for the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim staEqu As AeccStationEquations

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set staEqu = align.StationEquations

MsgBox "The number of StationEquations in this project is: " & staEqu.Count

StationIncrement Example

Sub Example_StationIncrement()

'This example returns the StationIncrement for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The StationIncrement for the first ProfileBlock in the collection is: " & alignProf.StationIncrement, vbInformation, "StationIncrement Example"

StationOffset Example

Sub Example_StationOffset()

- 'This example gets the StationOffset from user supplied point
- ' for the first Alignment in the collection..

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim returnPnt As Variant

Dim station As Double

Dim offset As Double

Dim direction As Double

' Get the point

returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point near the first alignment of the control of the contro

' Get the station offset location

align.StationOffset returnPnt(0), returnPnt(1), station, offset, direction

MsgBox "The StationOffset for the first Alignment is:" & vbCrLf & _

- " Station: " & station & vbCrLf & _
- " Offset: " & offset & vbCrLf & _
- " Direction: " & direction, vbInformation, "StationOffset Example"

Status Example

```
Sub Example_Status()
```

```
'This example displays the Status for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
```

Dim strStatus As String

```
' Set string to status of surface
Select Case surf.Status
Case kNoData
    strStatus = "has no data."
Case kNotBuilt
    strStatus = "is not built."
Case kUpToDate
    strStatus = "is up to date."
Case kOutOfDate
    strStatus = "is out of date."
End Select
```

MsgBox "The current surface " & strStatus, vbInformation, "Status Example"

SuperelevationCode Example

Sub Example_SuperelevationCode()

- 'This example returns the data for the first superelevation in the
- ' superelevation collection for the first alignment in the collection.

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim sElev As AeccSuperelevation

Set aligns = AeccApplication.ActiveProject.Alignments

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set sElev = align.Superelevations.Item(0)

' Get the alignment name Dim alignName As String alignName = align.Name

'Get the superelevation station and format it Dim station As String station = aligns.DoubleToStaFormat(sElev.station)

' Get the superelevation curve code

Dim crvCode As String

Select Case sElev.curveCode

Case kSERightHandCurve

crvCode = "Right Hand Curve"

Case kSELeftHandCurve

crvCode = "Left Hand Curve"

End Select

' Get the superelevation code

Dim supCode As String

Select Case sElev.SuperelevationCode

Case kSEFullCrown

supCode = "Full Crown"

Case kSEHalfCrown

```
supCode = "Half Crown"
Case kSECrownRemoved
supCode = "Crown Removed"
Case kSEFullSuperelevations
supCode = "Full Superelevation"
Case kSEReverseCurve
supCode = "Reverse Curve"
Case kSECompoundCurve
supCode = "Compound Curve"
End Select
```

```
MsgBox "The alignment name is: " & alignName & vbCrLf & _
"The data for the first superelevation is: " & vbCrLf & _
vbTab & "Station: " & station & vbCrLf & _
vbTab & "Curve Code: " & crvCode & vbCrLf & _
vbTab & "Superelevation Code: " & supCode, _
vbInformation, "SuperelevationCode Example"
```

Superelevations Example

Sub Example_Superelevations()

- 'This example returns the number of superelevations for the
- ' first alignment in the collection.

Dim align As AeccAlignment

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

MsgBox "The number of superelevations in first alignment is: " _ & align.Superelevations.Count , vbInformation, "Superelevations Example"

Surface Example

Sub Example_Surface()

- 'This example returns the PreferencesSurface object that
- ' is used to access the SurfaceLayer property

Dim surfPref As AeccPreferencesSurface

Set surfPref = AeccApplication.ActiveProject.preferences.Surface

MsgBox "The Surface preferences for SurfaceLayer is: " & surfPref.GetString(vbInformation, "Surface Example"

SurfaceName Example

Sub Example_SurfaceName()

- 'This example uses SurfaceName to displays the name of the surface
- ' for the first existing ground profile in the first alignment in the collection.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

MsgBox "The SurfaceName for the first existing ground profile is: " _ & EGProf.SurfaceName, vbInformation, "SurfaceName Example"

Surfaces Example

Sub Example_Surfaces()

- 'This example returns the vertical exaggeration factor for Surfaces
- ' Preferences in the current project.

Dim prefPrj As AeccPreferencesProject

Set prefPrj = AeccApplication.ActiveProject.Preferences

MsgBox "The vertical exaggeration factor for Surfaces Preferences in the curre, vbInformation, "Surfaces Example"

SymbolBlock Example

Sub Example_SymbolBlock()

- 'This example returns the SymbolBlock for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey SymbolBlock is " & dKey.SymbolBlo vbInformation, "SymbolBlock Example"

SymbolLayer Example

Sub Example_SymbolLayer()

- 'This example returns the SymbolLayer for the first
- 'DescriptionKey in the first DescriptionKeyFile in the collection.

Dim dKeyFile As AeccDescriptionKeyFile

Dim dKey As AeccDescriptionKey

Set dKeyFile = AeccApplication.ActiveProject.DescriptionKeyFiles.Item(0)

Set dKey = dKeyFile.Item(0)

MsgBox "The value for DescriptionKey SymbolLayer is " & dKey.SymbolLay vbInformation, "SymbolLayer Example"

SymbolManagerPath Example

Sub Example_SymbolManagerPath()

' This example returns the SymbolManagerPath setting. Dim prefFiles As AeccPreferencesFiles Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for SymbolManagerPath is: " & prefFiles.Symboll vbInformation, "SymbolManagerPath Example"

SystemPath Example

Sub Example_SystemPath()

' This example returns the SystemPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for SystemPath is: " & prefFiles.SystemPath, _ vbInformation, "SystemPath Example"

TangentLabelIncrement Example

Sub Example_TangentLabelIncrement()

'This example returns the TangentLabelIncrement for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The TangentLabelIncrement for the first ProfileBlock in the collectic & alignProf.TangentLabelIncrement, vbInformation, "TangentlabelIncrement

TangentLength Example

Examples:

1 AlignCurve

1 ParcelCurve

```
Sub Example_TangentLength_AlignCurve()
```

```
'This example returns the TangentLength for the first Curve found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Curve entity in the first Alignment."

'Find first Curve in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kCurve Then
alignMsg = "The TangentLength for the first Curve in the alignment is: "
& Format(alignEnt.TangentLength, "0.00")

Exit For
```

MsgBox alignMsg, vbInformation, "TangentLength Example"

End Sub

Next

End If

Sub Example_TangentLength_ParcelCurve()

- 'This example returns the TangentLength for the first curve in the
- ' first Parcel in the collection.

```
Dim parcel As AeccParcel
Dim parcelEnt As AeccParcelEntity
Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
Set parcelEnt = parcel.ParcelEntities.Item(0)

Dim parcelMsg As String
parcelMsg = "There is no Curve entity in the first Parcel."

' Find first Curve in the parcel
For Each parcelEnt In parcel.ParcelEntities
    If parcelEnt.Type = kParcelCurve Then
        parcelMsg = "The TangentLength for the first Curve in the Parcel is: " _
        & Format(parcelEnt.TangentLength, "0.00")
        Exit For
        End If
Next

MsgBox parcelMsg, vbInformation, "TangentLength Example"
```

TempPath Example

Sub Example_TempPath()

' This example returns the TempPath setting.
Dim prefFiles As AeccPreferencesFiles
Set prefFiles = AeccApplication.Preferences.Files

MsgBox "The current value for TempPath is: " & prefFiles.TempPath, vbInfor

TextAbove Example

Sub Example_TextAbove()

- 'This example returns the TextAbove setting for the
- ' first CurveText object in a selection set

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- 'Create the selection set based on a point selection
- ' and filter for CurveText objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AEC CURVETEXT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objCrvTxt As AeccCurveText Set objCrvTxt = ssetObj.Item(0)

MsgBox "The setting for TextAbove is: " & objCrvTxt.TextAbove , vbInforma

TextBelow Example

Sub Example_TextBelow()

- 'This example returns the TextBelow setting for the
- ' first CurveText object in a selection set

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- 'Create the selection set based on a point selection
- ' and filter for CurveText objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AEC CURVETEXT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objCrvTxt As AeccCurveText Set objCrvTxt = ssetObj.Item(0)

MsgBox "The setting for TextBelow is: " & objCrvTxt.TextBelow, vbInforma

TextHeight Example

Sub Example_TextHeight()

'This example displays the TextHeight setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The current value for TextHeight is: " & dbPref.TextHeight, _ vbInformation, "TextHeight Example"

TextLayer Example

Sub Example_TextLayer()

'This example returns the TextLayer for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The TextLayer for the first ProfileBlock in the collection is: " _ & alignProf.TextLayer, vbInformation, "TextLayer Example"

TextOffsetAbove Example

Sub Example_TextOffsetAbove()

- 'This example returns the TextOffsetAbove setting for the
- ' first CurveText object in a selection set

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- 'Create the selection set based on a point selection
- ' and filter for CurveText objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AEC CURVETEXT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objCrvTxt As AeccCurveText Set objCrvTxt = ssetObj.Item(0)

MsgBox "The setting for TextOffsetAbove is: " & objCrvTxt.TextOffsetAbove

TextOffsetBelow Example

Sub Example_TextOffsetBelow()

- 'This example returns the TextOffsetBelow setting for the
- ' first CurveText object in a selection set

On Error Resume Next

'Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- 'Create the selection set based on a point selection
- ' and filter for CurveText objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AEC CURVETEXT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objCrvTxt As AeccCurveText Set objCrvTxt = ssetObj.Item(0)

MsgBox "The setting for TextOffsetBelow is: " & objCrvTxt.TextOffsetBelow

TextSize Example

Sub Example_TextSize()

- 'This example returns the TextSize setting for the
- ' first CurveText object in a selection set

On Error Resume Next

' Delete existing SelectionSet ThisDrawing.SelectionSets("SSet").Delete

- ' Create the selection set based on a point selection
- ' and filter for CurveText objects

Dim ssetObj As AcadSelectionSet

Set ssetObj = ThisDrawing.SelectionSets.Add("SSet")

Dim gpCode(0) As Integer Dim dataValue(0) As Variant

gpCode(0) = 0
dataValue(0) = "AEC CURVETEXT"

Dim groupCode As Variant Dim dataCode As Variant

groupCode = gpCode
dataCode = dataValue
ssetObj.SelectOnScreen groupCode, dataCode

Dim objCrvTxt As AeccCurveText Set objCrvTxt = ssetObj.Item(0)

MsgBox "The setting for TextSize is: " & objCrvTxt.TextSize, vbInformation,

ThetaExt Example

```
Sub Example_ThetaExt()

' This example returns the ThetaExt for the first Spiral found in the
' first Alignment in the collection.
Dim align As AeccAlignment
Dim alignEnts As AeccAlignEntity
Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

' Find first Spiral in the alignment
For Each alignEnt In align.AlignEntities
    If alignEnt.Type = kSpiral Then
        alignMsg = "The ThetaExt for the first Spiral in the alignment is: " & alig Exit For
    End If
```

MsgBox alignMsg, vbInformation, "TheataExt Example"

End Sub

Next

Time Example

Sub Example_Time()

'This example returns the Time for the first FileLock in the collection.

Dim filelock As AeccFileLock

Set filelock = AeccApplication.ActiveProject.fileLocks.Item(0)

MsgBox "The Time of the first FileLock in the collection is: " & filelock.Time vbInformation, "Time Example"

TinPoints Example

Sub Example_TinPoints()

'This example returns the number of TinPoints in the first

Dim surf As AeccSurface

Dim surfOut As AeccSurfaceOutputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfOut = surf.Outputs

MsgBox "The number of TinPoints in the first Surface is: " & surfOut.TinPoints, vbInformation, "TinPoints Example"

^{&#}x27; surface in the collection.

TotalX Example

```
Sub Example_TotalX()
```

```
'This example returns the TotalX for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The TotalX for the first Spiral in the alignment is: " & alignExit For

End If

Next

MsgBox alignMsg, vbInformation, "TotalX Example"
```

TotalY Example

```
Sub Example_TotalY()
```

```
'This example returns the TotalY for the first Spiral found in the 'first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnts As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Dim alignMsg As String
alignMsg = "There is no Spiral entity in the first Alignment."

'Find first Spiral in the alignment

For Each alignEnt In align.AlignEntities

If alignEnt.Type = kSpiral Then

alignMsg = "The TotalY for the first Spiral in the alignment is: " & alignExit For
End If

Next

MsgBox alignMsg, vbInformation, "TotalY Example"
```

Type Example

Examples:

- 1 AlignEntity, AlignCurve, AlignCurve, AlignTangent
- 1 **Boundary**
- 1 Breakline
- 1 <u>CrossSectionSurface</u> (Civil Engineering Feature)
- 1 <u>EGProfile</u> (Civil Engineering Feature)
- 1 FGProfile (Civil Engineering Feature)
- 1 ParcelEntity, ParcelCurve, ParcelLine
- 1 StationEquation
- 1 Surface
- 1 Watershed

Sub Example_Type_AlignEntity()

- 'This example returns the Type setting for the first entity in the
- ' first Alignment in the collection.

Dim align As AeccAlignment

Dim alignEnt As AeccAlignEntity

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set alignEnt = align.AlignEntities.Item(0)

Dim alignType As String

Select Case alignEnt.Type Case kCurve alignType = "Curve"

```
Case kSpiral
    alignType = "Spiral"
  Case kTangent
    alignType = "Tangent"
  End Select
  MsgBox "The Type for the first entity in the alignment is: " & alignType _
    , vbInformation, "Type Example"
End Sub
Sub Example_Type_Boundary()
  'This example returns the Type
  ' for the first Boundary in the collection.
  Dim surf As AeccSurface
  Dim bound As AeccBoundary
  Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Set bound = surf.Inputs.Boundaries.Item(0)
  Dim bndType As String
  Select Case bound. Type
  Case kBoundaryTypeShow
    bndType = "Visible"
  Case kBoundaryTypeHide
    bndType = "Hidden"
  Case kBoundaryTypeOuter
    bndType = "Outer"
  End Select
  MsgBox "The Type for the first Boundary is: " & bndType _
    , vbInformation, "Type Example"
End Sub
```

Liid Odo

Sub Example_Type_BreakLine()

```
'This example returns the Type
```

Dim surf As AeccSurface

Dim brkLine As AeccBreakLine

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set brkLine = surf.Inputs.BreakLines.Item(0)

Dim brkType As String

Select Case brkLine. Type

Case kStandaed

brkType = "Standard"

Case kProximety

brkType = "Proximity"

Case kWallLeft

brkType = "Wall Left"

Case kWallRight

brkType = "Wall Right"

Case kNondestructive

brkType = "Non Destructive"

End Select

MsgBox "The Type for the first BreakLine is: " & brkType _ , vbInformation, "Type Example"

End Sub

Sub Example_Type_CrossSectionSurface()

Dim aligns As AeccAlignments

Dim align As AeccAlignment

Dim xSect As AeccCrossSection

Dim xSectSurf As AeccCrossSectionSurface

^{&#}x27; for the first BreakLine in the collection.

^{&#}x27;This example returns the surface type for the first cross

^{&#}x27; section surface in the collection for the first cross section

^{&#}x27; in the collection.

```
Set aligns = AeccApplication.ActiveProject.Alignments
Set align = aligns. Item(0)
Set xSect = align.CrossSections.Item(0)
Set xSectSurf = xSect.CrossSectionSurfaces.Item(0)
' Get the alignment name
Dim alignName As String
alignName = align.Name
'Get the cross section station and format it
Dim station As String
station = aligns.DoubleToStaFormat(xSect.station)
'Get the cross section surface name
Dim surfName As String
surfName = xSectSurf.Name
' Get the cross section surface type
Dim xSurfType As String
Select Case xSectSurf.Type
Case kExistingGround
  xSurfType = "Existing Ground"
Case kTopSurface
  xSurfType = "Top Surface"
Case kDatumSurface
  xSurfType = "Datum Surface"
Case kTemplateSurface
  xSurfType = "Template Surface"
Case kSubassemblySurface
  xSurfType = "Subassembly Surface"
Case kMatchSurface
  xSurfType = "Match Surface"
End Select
MsgBox "The alignment name is: " & alignName & vbCrLf & _
  "The first cross section is at station: " & station & vbCrLf & _
```

"The first surface name is: " & surfName & vbCrLf & _
"The first surface type is: " & xSurfType, vbInformation , "Type Example"

End Sub

Sub Example_Type_EGProfile()

'This example returns the Type setting for the first existing ground profile

' in the first alignment in the collection.

Dim align As AeccAlignment

Dim EGProf As AeccEGProfile

Set align = AeccApplication.ActiveProject.Alignments.Item(0)

Set EGProf = align.EGProfiles.Item(0)

Dim EGProfType As String

Select Case EGProf. Type

Case kEgCenter

EGProfType = "Center"

Case kEgLeft

EGProfType = "Left"

Case kEgNone

EGProfType = "None"

Case kEgRight

EGProfType = "Right"

End Select

MsgBox "The Type for the first existing ground profile in the alignment is: " & , vbInformation, "Type Example"

End Sub

Sub Example_Type_FGProfile()

Dim align As AeccAlignment

^{&#}x27;This example returns the Type setting for the finished ground profile

^{&#}x27; in the first alignment in the collection.

Dim FGProf As AeccFGProfile Set align = AeccApplication.ActiveProject.Alignments.Item(0) Set FGProf = align.FGProfiles.Item(0)

Dim FGProfType As String

```
Select Case FGProf.Type
Case kFgCenter
  FGProfType = "Center"
Case kFgDitchLeft
  FGProfType = "Ditch Left"
Case kFgDitchRight
  FGProfType = "Ditch Right"
Case kFgLeft1
  FGProfType = "Left 1"
Case kFgLeft2
  FGProfType = "Left 2"
Case kFgLeft3
  FGProfType = "Left 3"
Case kFgLeft4
  FGProfType = "Left 4"
Case kFgLeft5
  FGProfType = "Left 5"
Case kFgLeft6
  FGProfType = "Left 6"
Case kFgLeft7
  FGProfType = "Left 7"
Case kFgLeft8
  FGProfType = "Left 9"
Case kFgNone
  FGProfType = "None"
Case kFgRight1
  FGProfType = "Right 1"
Case kFgRight2
  FGProfType = "Right 2"
Case kFgRight3
  FGProfType = "Right 3"
```

```
Case kFgRight4
    FGProfType = "Right 4"
  Case kFgRight5
    FGProfType = "Right 5"
  Case kFgRight6
    FGProfType = "Right 6"
  Case kFgRight7
    FGProfType = "Right 7"
  Case kFgRight8
    FGProfType = "Right 8"
  End Select
  MsgBox "The Type for the finished ground profile in the alignment is: " & FG.
    , vbInformation, "Type Example"
End Sub
Sub Example_Type_ParcelEntity()
  'This example returns the Type for the first entity in the
  ' first Parcel in the collection.
  Dim parcel As AeccParcel
  Dim parcelEnt As AeccParcelEntity
  Set parcel = AeccApplication.ActiveProject.Parcels.Item(0)
  Set parcelEnt = parcel.ParcelEntities.Item(0)
  Dim parcelType As String
  Select Case parcelEnt.Type
  Case kParcelCurve
    parcelType = "Curve"
  Case kParcelLine
    parcelType = "Line"
  End Select
```

MsgBox "The Type of the first entity in the Parcel is: " & parcelType, _

vbInformation, "Type Example"

Case kUnknownSurface

Case kTin

surfType = "The surface type is Unknown."

surfType = "The surface type is Surface."

```
Sub Example_Type_StationEquation()
  'This example returns the Type setting for StationEquation.
  Dim align As AeccAlignment
  Dim staEqu As AeccStationEquation
  Set align = AeccApplication.ActiveProject.Alignments.Item(0)
  Set staEqu = align.StationEquations.Item(0)
  ' Convert the type to a string.
  Select Case staEqu.Type
  Case kIncreasing
    staType = "Station Equation increases stationing."
  Case kDecreasing
    staType = "Station Equation decreases stationing."
  End Select
  MsgBox "The current value for Type is " & staType, vbInformation, "Type Exa
End Sub
Sub Example_Type_Surface()
  'This example returns the Type for the first surface in the collection.
  Dim surf As AeccSurface
  Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)
  Dim surfType as String
  'Convert the type to a string.
  Select Case surf. Type
```

```
Case kCompositeVolumn
surfType = "The surface type is Composite Volume."
Case kGridVolumn
surfType = "The surface type is Grid Volume."
End Select
```

MsgBox surfType, vbInformation, "Type Example"

End Sub

```
Sub Example_Type_Watershed()
```

```
' This example returns the Type or Watershed
' for the first WaterShed in the collection.

Dim surf As AeccSurface
```

Dim wShed As AeccWaterShed

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set wShed = surf.Outputs.WaterSheds.Item(0)

Dim wShedType As String

```
Select Case wShed.Type

Case kBoundaryPoint

wShedType = "includes a boundary point."

Case kBoundarySegment

wShedType = "includes part of a boundary."

Case kLocalMin

wShedType = "is a depression."

Case kFlatArea

wShedType = "has a flat area."

Case kMultiDrain

wShedType = "has multiple drains."

End Select
```

MsgBox "The first WaterShed" & wShedType, vbInformation, "Type Example

UnLockPoints Example

Sub Example_UnLockPoints()

```
'This examples locks a series of point numbers and
'then unlocks a subset of those point numbers.

Dim cogoPnts As AeccCogoPoints

Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

'Create locked point string

Dim pntString As String

pntString = "1,4-8,40,10,12"
```

'Lock points cogoPnts.LockPoints (pntString)

MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _ vbInformation, "UnLockPoints Example"

' Create unlock point string pntString = "4-8"

' Unlock points cogoPnts.UnlockPoints (pntString)

MsgBox "Locked point numbers are " & cogoPnts.LockedPointNumbers, _ vbInformation, "UnLockPoints Example"

UpperRight Example

Sub Example_UpperRight()

'This example returns the Datum Elevation for the first ProfileBlock

' in the collection.

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

Dim coords As Variant

coords = alignProf.UpperRight

MsgBox "The UpperRight coordinates for the first Alignment Profile are:" & v

"X Value: " & Format(coords(0), "0.00") & vbCrLf & _

"Y Value: " & Format(coords(1), "0.00"), vbInformation, "UpperRight Exan

UsedPointNumbers Example

Sub Example_UsedPointNumbers()

' This gets the UsedPointNumbers in the CogoPoints collection Dim cogoPnts As AeccCogoPoints Set cogoPnts = AeccApplication.ActiveProject.CogoPoints

MsgBox "The UsedPointNumbers in the CogoPoints collection are: " & cogoP vbInformation, "UsedPointNumbers Example"

User Example

Sub Example_User()

- 'This example returns the current setting of LastUsedProject
- ' from the preferences object.

Dim prefs As AeccPreferences

Set prefs = AeccApplication.Preferences

MsgBox "The last used project was " & prefs.User.LastUsedProj, vbInformatic

Utility Example

Sub Example_Utility()

'This example returns the AngleToString value for 0.785398 radians using

' the active document.

Dim doc As AeccDocument

Set doc = AeccApplication.ActiveDocument

MsgBox "The setting for AngleToString is " & doc.Utility.AngleToString(0.78 , vbInformation, "Utility Example"

VerticalGridLayer Example

Sub Example_VerticalGridLayer()

'This example returns the VerticalGridLayer for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The VerticalGridLayer for the first ProfileBlock in the collection is:
& alignProf.VerticalGridLayer, vbInformation, "VerticalGridLayer Example

VerticalScale Example

Examples:

- 1 <u>CrossSectionBlock</u> (Civil Engineering Feature)
- 1 DatabasePreferences
- 1 ProfileBlock (Civil Engineering Feature)

Sub Example_VerticalScale_CrossSectionBlock()

- 'This example returns the vertical scale for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

Set alignXSects = AeccApplication.ActiveDocument.CrossSectionBlocks
Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The vertical scale for the alignment cross section at station " & static Format(alignXSect.VerticalScale, "0.00"), vbInformation, "VerticalScale Example 2009 and Static Example 2009 and Stat

End Sub

Sub Example_VerticalScale_DatabasePreferences()

'This example returns the vertical scale setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.Preferences

MsgBox "The current vertical scale for this drawing is: " & dbPref.VerticalScale vbInformation, "VerticalScale Example"

End Sub

Sub Example_VerticalScale_ProfileBlock()

'This example returns the vertical scale for the first alignment profile

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

MsgBox "The vertical scale for the first alignment profile is: " & _ Format(alignProf.VerticalScale, "0.00"), vbInformation, "VerticalScale Exan

^{&#}x27; in the collection.

Volume Example

Sub Example_Volume()

'This example returns the Volume for the first surface in the collection.

Dim surf As AeccSurface

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

MsgBox "The Volume for the first surface is: " & surf. Volume, vbInformation,

VolumnDisplay Example

Sub Example_VolumeDisplayUnit()

```
'This example returns the VolumeDisplayUnit setting for the current drawing. Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences
```

'Convert the volume display unit to a string.

Dim unit As String

Select Case dbPref.VolumeDisplayUnit

Case aecUnitCubicInch

unit = "cubic inch"

Case aecUnitCubicFoot

unit = "cubic foot"

Case aecUnitCubicYard

unit = "cubic yard"

Case aecUnitCubicMil

unit = "cubic millimeters"

Case aecUnitCubicCentimeter

unit = "cubic centimeters"

Case aecUnitCubicDecimeter

unit = "cubic decimeters"

Case aecUnitCubicMeter

unit = "cubic meters"

End Select

MsgBox "The current value for VolumeDisplayUnit is " & unit, _ vbInformation, "VolumnDisplayUnit Example"

VolumnPrecision Example

Sub Example_VolumePrecision()

'This example displays the VolumePrecision setting for the current drawing. Dim dbPref As AeccDatabasePreferences
Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The current value for VolumePrecision is: " & dbPref.VolumePrecisi vbInformation, "VolumnPrecision Example"

VolumnSuffix Example

Sub Example_VolumeSuffix()

'This example displays the VolumeSuffix setting for the current drawing.

Dim dbPref As AeccDatabasePreferences

Set dbPref = AeccApplication.ActiveDocument.preferences

MsgBox "The current value for VolumeSuffix is " & dbPref.VolumeSuffix, _ vbInformation, "VolumnSuffix Example"

Watersheds Example

Sub Example_Watersheds()

- 'This example returns the number of WaterSheds in the first
- ' surface in the collection.

Dim surf As AeccSurface

Dim surfOut As AeccSurfaceOutputs

Set surf = AeccApplication.ActiveProject.Surfaces.Item(0)

Set surfOut = surf.Outputs

MsgBox "The number of WaterSheds in the first Surface is: " & surfOut.Water , vbInformation, "WaterSheds Example"

XyToEastNorth Example

```
Sub Example_XyToEastNorth()
```

```
'This example returns the Easting and Northing for an AutoCAD XY.
Dim util As AeccUtility
Set util = AeccApplication.ActiveDocument.Utility
Dim ptXY (0 To 2) As Double
Dim ptEN(0 To 2) As Double
Dim var1 As Variant
Dim str As String
ptXY(0) = 1000
ptXY(1) = 1000
ptXY(2) = 0
'Convert point to Easting, Northing
var1 = util.XyToEastNorth(ptXY)
ptEN(0) = var1(0)
ptEN(1) = var1(1)
ptEN(2) = var1(2)
' Make display string
str = Format(ptEN(0), "0.00") + ", " + Format(ptEN(1), "0.00") +
MsgBox "Point: " & str, vbInformation, "XyToEastNorth Example"
```

XyToOffsetElevation Example

Sub Example_XyToOffsetElevation()

- ' This example returns the offset and elevation for a selected point in the first
- ' alignment crosssection in the collection

Dim alignXSect As AeccCrossSectionBlock

Set alignXSect = AeccApplication.ActiveDocument.CrossSectionBlocks.Item(

Dim station As String Dim offElev As Variant Dim returnPnt As Variant

'Get the station for the first alignment cross section in the collection. station = alignXSect.station

'Get a point in the alignment cross section.
returnPnt = ThisDrawing.Utility.GetPoint(, vbCrLf & "Specify a point in the al

offElev = alignXSect.XyToOffsetElevation(returnPnt)

MsgBox "The station for the selected point is: " & Format(offElev(0), "0.00") a "The offset for the selected point is: " & Format(offElev(1), "0.00") _ , vbInformation, "XyToOffsetElevation Example"

XyToStationElevation Example

Sub Example_XyToStationElevation()

'This example returns the XyToStationElevation for the first ProfileBlock

' in the collection

Dim alignProf As AeccProfileBlock

Set alignProf = AeccApplication.ActiveDocument.ProfileBlocks.Item(0)

Dim staElev As Variant
Dim returnPnt As Variant

' Get a point in the alignment profile.

returnPnt = ThisDrawing.Utility.GetPoint(, "Enter a point in the alignment prot

staElev = alignProf.XyToStationElevation(returnPnt)

 $\label{lem:matter} MsgBox "The Station for the X value is: " \& Format(staElev(0), "0.00") \& vb("The Elevation for the Y Value is: " \& Format(staElev(1), "0.00") _ , vbInformation, "XyToStationElevation Example"$

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

<u>Status</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

FindPoint

GetElevation

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>SearchType</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

 $\underline{SetBoundingBox}$

Build

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

FindPoint

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Understanding Parcels

Understanding Profiles

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Collections

<u>AlignEntities</u>
ElevationContours

<u>Alignments</u> Faces

Boundaries FGProfiles (Civil Engineering

<u>BreakLines</u> Feature)

<u>CogoPoints</u> <u>FileLocks</u>

ContourItems <u>Parcels</u>

CrossSectionBlocks (Civil Engineering ParcelEntities

Feature)

CrossSectionPointCodes (Civil

Engineering Feature) PointGroups

<u>CrossSections</u> (Civil Engineering <u>PointGroupNames</u>

Feature)

<u>CrossSectionSurfaces</u> (Civil Feature)

Engineering Feature) Projects

<u>DEMFiles</u>

PVIs (Civil Engineering Feature)

PointFiles

ProfileBlocks (Civil Engineering

<u>DescriptionKeyFile</u>

<u>Prototypes</u>

<u>DescriptionKeyFiles</u>
<u>StationEquations</u>

<u>Documents</u> <u>SuperElevations</u> (Civil

<u>Drawings</u> Engineering Feature)

<u>Surfaces</u>

<u>TinPoints</u>
<u>EGProfiles</u> (Civil Engineering Feature)

<u>WaterSheds</u>

ElevationContours

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>LockedPointNumbers</u>

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

<u>Save</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

<u>AutoSave</u>

<u>Open</u>

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

<u>GetBoundingBox</u>

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>LockedPointNumbers</u>

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Parcels

See Also
Methods and Properties:
<u>Area3D</u>
<u>AverageGrade</u>

MaxFaceArea

MaxGrade

MeanElevation

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Understanding Surfaces

See Also
Methods and Properties:
<u>Area2D</u>
<u>AverageGrade</u>

MaxFaceArea

MaxGrade

MeanElevation

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>Save</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>Save</u>

ActiveX and VBA Developer's Guide:

<u>Understanding Alignments</u>

Understanding CogoPoints

See Also
Methods and Properties:
<u>Area2D</u>
Area3D

<u>MaxGrade</u>

MeanElevation

MaxFaceArea

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

CenterlineOffset Example

Sub Example_CenterlineOffset()

- 'This example returns the centerline offset for the
- ' first alignment cross section in the collection.

Dim alignXSects As AeccCrossSectionBlocks

Dim alignXSect As AeccCrossSectionBlock

 ${\bf Set\ align XSects} = {\bf Aecc Application. Active Document. Cross Section Blocks}$

Set alignXSect = alignXSects.Item(0)

'Get the station for the first alignment cross section in the collection Dim station As String station = alignXSect.station

MsgBox "The centerline offset for the alignment cross section at station " & station Tormat(alignXSect.CenterlineOffset, "0.00"), vbInformation, "Height Example of the alignment cross section at station to the station of the alignment cross section at station to the station of the alignment cross section at station to the alignment cross section at station at s

End Sub

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Understanding COM Wrappers

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

<u>Understanding the ActiveX Object Model</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

Name

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Understanding COM Wrappers

Understanding Cross Sections and Superelevations

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Understanding COM Wrappers

Understanding Cross Sections and Superelevations

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>PointNameSize</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Parcels

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

LockPoints

<u>UnlockPoints</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Understanding Project Management

Methods and Properties:

SpiralType2

ActiveX and VBA Developer's Guide:

<u>Understanding Alignments</u>

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

See Also
Methods and Properties:
Area2D
<u>Area3D</u>

<u>AverageGrade</u>

<u>MaxGrade</u>

MeanElevation

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

See Also
Methods and Properties:
Area2D
<u>Area3D</u>

<u>AverageGrade</u>

MaxFaceArea

MeanElevation

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

See Also
Methods and Properties
<u>Area2D</u>
<u>Area3D</u>

<u>AverageGrade</u>

MaxFaceArea

MaxGrade

<u>MinFaceArea</u>

MinGrade

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

See Also
Methods and Properties
Area2D
<u>Area3D</u>

<u>AverageGrade</u>

MaxFaceArea

MaxGrade

MeanElevation

<u>MinGrade</u>

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

See Also
Methods and Properties
<u>Area2D</u>
<u>Area3D</u>

<u>AverageGrade</u>

MaxFaceArea

MaxGrade

MeanElevation

<u>MinFaceArea</u>

NumberOfFaces

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

Description

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Understanding COM Wrappers

Understanding Cross Sections and Superelevations

Understanding Parcels

Understanding Profiles

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Understanding COM Wrappers

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding CogoPoints

Understanding COM Wrappers

See Also
Methods and Properties:
Area2D
<u>Area3D</u>

<u>AverageGrade</u>

MaxFaceArea

MaxGrade

MeanElevation

<u>MinFaceArea</u>

MinGrade

<u>Volume</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>NumberOfFaces</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Parcels Example

Sub Example_Parcels()

' This example returns the number of Parcels in the current project. Dim proj As AeccProject

Set proj = AeccApplication.ActiveProject

MsgBox "The number of Parcels in the current Project is: " & proj.Parcels.Cou , vbInformation, "Parcels Example"

End Sub

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>IsNameSupported</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

PointTolerance

ActiveX and VBA Developer's Guide:

Methods and Properties:

<u>PointOnLineTolerance</u>

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

FindPath

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

SpiralType2

ActiveX and VBA Developer's Guide:

<u>Understanding Alignments</u>

Methods and Properties:

LOffset

SpiralType1

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

Build

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Project Management

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Profiles

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding CogoPoints

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Alignments

Understanding Cross Sections and Superelevations

Understanding Profiles

See Also
Methods and Properties:

Area2D

Area3D

<u>AverageGrade</u>

MaxGrade

MaxFaceArea

MeanElevation

MinFaceArea

MinGrade

NumberOfFaces

ActiveX and VBA Developer's Guide:

Understanding Surfaces

Methods and Properties:

ActiveX and VBA Developer's Guide:

Understanding Surfaces