#### Sage 300 ERP (ACCPAC) COM API

#### **ACCPAC COM API Object Model**

The ACCPAC COM API is a COM library which provides access to Sage 300 ERP System Manager functions. The ACCPAC COM API is organized as a group of COM objects whose properties and methods access underlying System Manager functionality.

To access objects in ACCPAC COM API, you need to create and open an <u>AccpacSession</u> object. This object represents an authenticated session with System Manager.

#### **Quick Start**

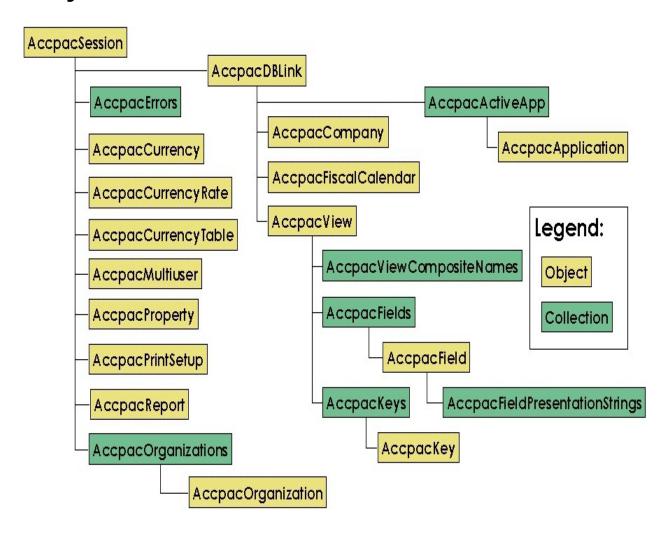
An application uses the Session class to establish a session with the Sage 300 ERP System Manager.

- 1. Creates an instance of the AccpacSession class.
- 2. Initializes the AccpacSession object by calling Init.
- 3. Opens the session by calling Open on the AccpacSession object, supplying a valid username and password.

Before accessing application data, an application needs to open a database connection to the intended company database. This can be done by obtaining a <a href="AccpacDBLink">AccpacDBLink</a> object from the <a href="AccpacSession">AccpacSession</a> object's <a href="OpenDBLink">OpenDBLink</a> method.

Applications can access ACCPAC application features and data through AccpacView objects. These object represent ACCPAC application views, which provide access to application data and business logic. AccpacView objects can be obtained from the AccpacDBLink object's OpenView method.

# Sage 300 ERP (ACCPAC) COM API Object Model



#### tagBlockDateMatchEnum

Specifies the date matching method used to determine whether a currency belongs to a currency block.

Name	Value	Description
BLOCK_EXACT_DATE	1	Match exact date only.
BLOCK_ON_OR_BEFORE	2	Match on or before the specified date.

### tagCompanyHandleInactiveGLAccounts

Specifies how to handle inactive General Ledger accounts.

Name	Value	Description
INACTIVE_GLACCOUNTS_IGNORE	0	Ignore.
INACTIVE_GLACCOUNTS_WARNING	1	Display a warning.
INACTIVE_GLACCOUNTS_ERROR	2	Raise an error.

### tagCompanyHandleLockedFscPeriods

Specifies how to handle locked fiscal periods.

Name	Value	Description
LOCKED_FSCPERIODS_IGNORE	0	Ignore.
LOCKED_FSCPERIODS_WARNING	1	Display a warning.
LOCKED_FSCPERIODS_ERROR	2	Raise an error.

### tagCompanyHandleNonexistantGLAccou

Specifies how to handle non-existent General Ledger accounts.

Name	Value	Description
NONEXISTANT_GLACCOUNTS_IGNORE	0	Ignore.
NONEXISTANT_GLACCOUNTS_WARNING	1	Display a warning.
NONEXISTANT_GLACCOUNTS_ERROR	2	Raise an error.

### tagCurrencySymbolDisplayEnum

Specifies how to display the currency symbol with a currency amount.

Name	Value	Description
BEFORE_WITH_SPACE	1	Display currency symbol before the amount, with a space in between.
BEFORE_WITHOUT_SPACE	2	Display currency symbol before the amount, without spaces.
AFTER_WITH_SPACE	3	Display currency symbol after the amount, with a space in between.
AFTER_WITHOUT_SPACE	4	Display currency symbol after the amount, without spaces.

### tagDatabaseSeries

Specifies the database series of the current ACCPAC System Manager installation.

Name	Value	Description
DB_PERVASIVE	0	Pervasive.SQL
DB_SQLSERVER	2	Microsoft SQL Server
DB_DB2	3	IBM DB2
DB_ORACLE	4	Oracle
DB_INGRES	5	CA-Ingres

#### tagDateMatchEnum

Specifies the date matching method to use when determining an exchange rate with respect to a given date.

Name	Value	Description
EXACT	1	Gets the rate for the same date as the given date.
LATER	2	Gets the rate for the first date after the given date.
EARLIER	3	Gets the rate for the first date before the given date.

### tagDBLinkFlagsEnum

Specifies the access (read/write) mode of a database link or connection.

Name	Value	Description
DBLINK_FLG_READWRITE	0	Read-write database connection.
DBLINK_FLG_READONLY	1	Read-only database connection.
DBLINK_FLG_READUNCOMMITTED		Read-only database connection, but it will read uncommitted values (for Microsoft SQL only).

### tagDBLinkTypeEnum

Specifies the ACCPAC database type.

Name	Value	Description
DBLINK_SYSTEM	0	Link or connection to a system database.
DBLINK_COMPANY	1	Link or connection to a company database.

### tagDistFileType

Specifies the format of a file path.

Name	Value	Description
DF_DYNAMIC	0	Format is dynamic and is dependent upon the client's current session connection. If the connection is remote, it uses the URL as the file path. If the connection is local, it uses the local file path.
DF_LOCAL	1	Use local file path.
DF_URL	2	Use URL as the file path.

### tagErrorPriority

Specifies the type of an application error or message.

Name	Value	Description
ERRPRI_SEVERROR	0	Severe error
ERRPRI_MESSAGE	1	Message
ERRPRI_WARNING	2	Warning
ERRPRI_ERROR	3	Error
ERRPRI_SECURITY	4	Security violation

### tagEventReason

Specifies the reason an event was fired.

Name	Value	Description
RSN_INIT	0	Init, RecordClear, or RecordGenerate was called.
RSN_ADDNEW	1	Insert or RecordGenerate was called.
RSN_DELETE	2	Delete was called.
RSN_FETCH	3	Fetch or FilterFetch was called.
RSN_FETCHLOCK	4	FetchLock or FilterFetch was called.
RSN_GOTOP	5	GoTop was called.
RSN_GOBOTTOM	6	GoBottom was called.
RSN_FIELDCHANGE	7	Field value has changed.
RSN_READ	8	Read was called.
RSN_GONEXT	9	GoNext was called.
RSN_BLKPUT	10	BlkPut was called.
RSN_GOTOBOOKMARK	11	GotoBookMark or GotoRecordNumber was called.
RSN_READLOCK	12	ReadLock was called.

RSN_GOPREV	13	GoPrev was called.
RSN_UPDATE	14	Update was called.
RSN_REFRESH	15	Field values might have changed. Event handlers should refresh the values they are interested in.
RSN_CANCEL	16	Cancel or RevisionCancel was called.
RSN_REFRESH_RECNOTFOUND	17 (&H11)	Navigation methods couldn't scroll to another record.
RSN_POST	18 (&H12)	Post or RevisionPost was called.
RSN_FILTERDELETE	19 (&H13)	FilterDelete was called.
RSN_TABLEEMPTY	20 (&H14)	TableEmpty was called.
RSN_PROCESS	21 (&H15)	Process was called.

### tagEventStatus

Specifies the status of an event.

Name	Value	Description
STATUS_OK	0	Proceed with event.
STATUS_CANCEL	1	Event should be cancelled.
STATUS_CANTDENY	2	Event will proceed and cannot be cancelled.

### tag Field Attribute Enum

Specifies various field attributes.

Name	Value	Description
FLD_CHANGED	1	Field value has been changed.
FLD_ENABLED	2	Field is enabled and accessible.
FLD_EDITABLE	4	Field value can be edited.
FLD_KEY	8	Field is a segment of the current key for the view.
FLD_CALCULATE	16	Field's value has been calculated from other fields.
FLD_TYPE	32	Field's type, size, or precision may change.
FLD_PRESENTATION	64	Field presentation mask or list (strings) may change.
FLD_REQUIRED	128	Field value is required for inserting or updating a record.
FLD_CHECKEDITABLE	256	Field's editable attribute (FLD_EDITABLE) may change.

# tag Field Presentation Type Enum

Specifies the type of presentation information defined for a field.

Name	Value	Description
FLD_PRESENTS_NONE	32	Field does not have presentation information.
FLD_PRESENTS_LIST	76	Field has a presentation list.
FLD_PRESENTS_MASK	77	Field uses a presentation mask to control the display format.

# tagFieldTypeEnum

Specifies the data type of a field.

Name	Value	Description	Stored in database as
FLD_CHAR	1	String	ASCII (blank padded)
FLD_BYTE	2	String	Binary (array of bytes)
FLD_DATE	3	Date	Decimal
FLD_TIME	4	Date	Decimal
FLD_REAL	5	Double	8-byte IEEE long real
FLD_DECIMAL	6	Decimal	BCD
FLD_INT	7	Integer	16-bit signed integer
FLD_LONG	8	Long	32-bit signed integer
FLD_BOOL	9	Boolean	16-bit signed integer (0=false, non-zero=true)

### tagFileLocation

Specifies the location of a particular file. This is used in routines that involve file transfer between server and client machines and is used to indicate the file location.

Name	Value	Description
LOC_CLIENT		The specified file path is a local path on the client machine.
LOC_COMPANY		Specifies the COMPANY subdirectory under the Shared Data Directory on the server.

### tagFilterOriginEnum

Specifies the origin from where the filter should be applied.

Name	Value	Description
FILTERSELECT_FROMSTART	1	The filter should be applied to all records in the view. The current value of the view's current key is disregarded.
FILTERSELECT_FROMCURRENT	2	The filter should be applied only to the current record and all records after it.

### tagFilterStrictnessEnum

Specifies the behaviour of a filter used for record deleting under questionable situations related to referential integrity.

Name	Value	Description
FILTERDELETE_STRICT	1	Applies the filter to delete records only if it does not violate referential integrity. Otherwise, return an error.
FILTERDELETE_TRY	2	Applies the filter to delete records if possible, without considering if it results in an orphaned secondary table and detail records.
FILTERDELETE_SIMULATE	3	Applies the filter to delete records using the Strict method. If that results in a error, fallback to Fetch-Delete calls to the view.

### tagLicenseStatus

Indicates the status of an application license.

Name	Value	Description
LICENSE_OK	0	The license is valid.
LICENSE_NOTFOUND	-1 (&HFFFFFFFF)	License of the specified application could not be found.
LICENSE_EXPIRED	-2 (&HFFFFFFE)	License of the specified application has expired.

### tagMultiuserStatus

Return code of Multiuser routines (where resource locking may be involved).

Name	Value	Description
MUSTATUS_SUCCESS	0	The requested call succeeded.
MUSTATUS_LOCKED	1	The specified resource has already been locked by another process.
MUSTATUS_NOT_LOCKED	2	The specified resource was not previously locked.

### tagNegativeDisplayEnum

Specifies how negative currency amounts should be displayed.

Name	Value	Description
TRALING_MINUS	1	Display with a negative (minus) sign after the amount.
LEADING_MINUS	2	Display with a negative (minus) sign before the amount.
BRACKETS	3	Enclose the amount in brackets ().

### tagOperationType

Specifies whether to perform an import or export operation.

Name	Value	Description
T_IMPORT	0	Import operation.
T_EXPORT	1	Export operation.

### tag Organization Type Enum

Specifies the type of an ACCPAC database.

Name	Value	Description
ORG_SYSTEM	0	System database
ORG_COMPANY	1	Company database
ORG_COMBINED	2	Reserved

### tagOutputDestinationEnum

Specifies whether the report should be generated on the client (locally) or on the remote ACCPAC server.

Name	Value	Description
OD_SERVER	1	Generate the report on the remote ACCPAC server.
OD_CLIENT	2	Generate the report on the local machine (client).

### tagPeriodType

Specifies the fiscal period definition for the current company.

Name	Value	Description
PERTYPE_7DAYS	1	A fiscal period contains 7 days.
PERTYPE_WEEKLY	2	A fiscal period represents a week, respecting the Sunday/Saturday rule.
PERTYPE_MONTHLY	3	A fiscal period represents a month.

# tagPrintDestinationEnum

Specifies the destination where reports should be printed.

Name	Value	Description
PD_PRINTER	1	Print to printer.
PD_FILE	2	Output to a file.
PD_HTML	3	Output to an HTML file.
PD_PRINTCONF	4	Print to printer to check paper alignment. Only the first page of the report is printed.
PD_PREVIEW	5	Preview report in a preview window.
PD_EMAIL	6	Send an email with the print output as an attachment.
PD_SCHEDULE	7	Outputs to Crystal Info.
PD_MSGFAX	8	Fax output using ACCPAC Messenger, if installed.

### tagPrintFormatEnum

Indicates the desired print output format. This is only applicable if the print destination is a file.

Name	Value	Description
PF_NONE	0	Not specified. The output format will be determined according to the user's preference.
PF_PDF	1	Adobe Acrobat format.
PF_RTF	2	Rich text format.

### tagProductSeries

Specifies the ACCPAC product series for the current ACCPAC System Manager installation.

Name	Value	Description
PRODUCT_ENTERPRISE	1	Enterprise Edition
PRODUCT_CGA	2	CGA Edition
PRODUCT_CORPORATE	4	Corporate Edition
PRODUCT_DISCOVERY	5	Discovery Edition
PRODUCT_SMALLBUSINESS	6	Small Business Edition

### tagPropertyType

Specifies the data type desired when retrieving an ACCPAC property.

Name	Value	Description
PROP_AS_ARRAY	1	The property should be returned as an array of bytes.
PROP_AS_STRING	2	The property should be returned as a string.

### tagRateOperatorEnum

Specifies the operator that should be used when performing currency exchange calculations.

Name	Value	Description
MULTIPLICATION	1	The home (functional) currency amount should be obtained by multiplying the source currency amount by the exchange rate.
DIVISION		The home (functional) currency amount should be obtained by dividing the source currency amount by the exchange rate.

### tagReferentialIntegrityEnum

Specifies the referential integrity flag of a view.

Name	Value	Description
VIEW_REFINT_CASCADE		Changes to the view should be cascaded to maintain referential integrity.

# tagReportOutputType

Specifies the output type of a report file.

Name	Value	Description
REPORT_PDF	1	Adobe Acrobat format.
REPORT_RTF	2	Rich text format.

# tagRotoViewTypeEnum

Specifies the type of a view.

Name	Value	Description
ROTO_VIEW	1	View is a normal (base) view.
ROTO_VIEW_SUBCLASS	2	View is a subclass view.
ROTO_VIEW_STUB	3	View is a stub view.

# tagStatusReportingType

Name	Value	Description
REP_NT_SIMPLE	0	
REP_NT_RETURNSTAT	1	
REP_NT_DLL	2	

# tagSystemAccessEnum

Specifies the current system access mode of a view.

Name	Value	Description
VIEW_SYSACCS_NORMAL	0	Normal view access.
VIEW_SYSACCS_IMPORT	1	View is in import mode.
VIEW_SYSACCS_EXPORT	2	View is in export mode.
VIEW_SYSACCS_INTEGRITY_CHECK	3	View is being accessed by an integrity checker.
VIEW_SYSACCS_MACRO	4	View is being accessed by a macro.
VIEW_SYSACCS_ACTIVATION	5	View is being accessed by an activation object.
VIEW_SYSACCS_CONVERSION	6	View is in data conversion mode.
VIEW_SYSACCS_POSTING	10	View is being accessed by a posting object.

# tagViewReturnCode

Specifies the view operation's return code.

Name	Value	Description
Original Errors		
VC_SUCCESS	0	Operation succeeded
VC_LOAD_FAILED	100 (&H64)	View load failed
VC_OPEN_FAILED	101 (&H65)	View open failed
VC_COMPOSE_FAILED	102 (&H66)	View compos failed
VC_ROTOENTRY_FAILED	103 (&H67)	Caller has requested an operation not implemented by this view
VC_OPEN_FAILED_PS	500 (&H1F4)	View open failed due to process serve error
VC_OPEN_FAILED_PSCONNECT	501 (&H1F5)	View open failed due to process serve connection error

Old (Redundant) Errors		
VC_OLD_GENERAL	1	(OLD) Unspecified failure
VC_OLD_WARNING	10	(OLD) Unspecified warning
VC_OLD_OTHER	2	(OLD) Unspecified failure
VC_OLD_RECORD_NOT_FOUND	1	(OLD) Specified record doesn' exist in table
VC_OLD_RECORD_NO_MORE_DATA	1	(OLD) End of filtered table encountered
VC_OLD_RECORD_EXISTS	1	(OLD) Record already exists
VC_OLD_RECORD_DUPLICATE	1	(OLD) Insert has found tha the record already exists
VC_OLD_TABLE_EXISTS	1	(OLD) Create has found tha the table already exists
Extended Errors		
VC_GENERAL	1000 (&H3E8)	Unspecified failure
	I	I

VC_RECORD_NOT_FOUND	1020 (&H3FC)	Specified record doesn' exist in table
VC_RECORD_NO_MORE_DATA	1021 (&H3FD)	End of filtered table encountered
VC_RECORD_EXISTS	1022 (&H3FE)	Record already exists
VC_RECORD_DUPLICATE	1023 (&H3FF)	Insert has found that the record already exists
VC_RECORD_INVALID	1024 (&H400)	Record level validation failed
VC_RECORD_LOCKED	1025 (&H401)	Record is locked by another user or task
VC_RECORD_CONFLICT	1026 (&H402)	Record updated by another user or task
VC_RECORD_NOT_LOCKED	1027 (&H403)	Caller has requested an operation which requires a locked record
VC_TABLE_EXISTS	1040 (&H410)	Create has found that the table already exists

VC_TABLE_NOT_FOUND	1041 (&H411)	Table that is required by th operation doesn't exist
VC_PERMISSION_NONE	1060 (&H424)	User doesn't have security to do this view operation
VC_MEMORY_NO_MORE	1080 (&H438)	Out of memor
VC_MEMORY_BUFFER_LIMIT	1081 (&H439)	The size of the buffer supplied doesn't meet the operation' required size
VC_FILTER_SYNTAX	1100 (&H44C)	Filter's syntax couldn't be parsed
VC_FILTER_OTHER	1101 (&H44D)	Unspecified filter error
VC_KEY_INVALID	1120 (&H460)	Record's key validation failed
VC_KEY_NUMBER	1121 (&H461)	Supplied key number is invalid
VC_KEY_CHANGED	1122 (&H462)	Attempt to change alternate key field values budictionary doesn't permithem to be

		modified
VC_FIELD_INVALID	1140 (&H474)	Field validatio failed
VC_FIELD_NUMBER	1141 (&H475)	Field number doesn't exist i view
VC_FIELD_INDEX	1142 (&H476)	Field index doesn't exist i view
VC_FIELD_DISABLED	1143 (&H477)	Attempt to access disabled field
VC_FIELD_READONLY	1144 (&H478)	Attempt to change read-only field
VC_TRANSACTION_NONE	1160 (&H488)	A transaction is required to do the operation
VC_TRANSACTION_OPEN	1161 (&H489)	Transaction couldn't be opened to do the operation
VC_REVISION_PROTOCOL	1180 (&H49C)	Invalid revisio API state or call
VC_DATABASE_PARAMETER	1200 (&H4B0)	Invalid link, table handle, or other parameter to database API

VC_DATABASE_LIMIT	1201 (&H4B1)	Database resources exceeded
VC_DATABASE_OTHER	1202 (&H4B2)	Unspecified database erro
VC_DATABASE_DICTIONARY	1203 (&H4B3)	Database dictionary inconsistent
VC_RPC_FAILURE	1220 (&H4C4)	Remote procedure cal had communicatio failure
VC_OPERATION_CANCELLED	2000 (&H7D0)	The operation was cancelled
VC_NON_VIEW_ERROR	2001 (&H7D1)	Non-view errc
VC_APPLICATION_DEFINED_BASE	9000 (&H2328)	Base (lower limit) for application- defined error codes
VC_APPLICATION_DEFINED_END	9999 (&H270F)	End (upper limit) of application- defined error code range
Extended Warnings	_	
VC_WARNING	-1 (&HFFFFFFFF)	Normal warning
VC_WARNING_GENERAL	-1	Normal

	(&HFFFFFFF)	warning
VC_WARNING_APPLICATION_DEFINED_BASE	-1999 (&HFFFFF831)	Base (lower limit) for application- defined warning code:
VC_WARNING_APPLICATION_DEFINED_END	-1000 (&HFFFFFC18)	End (upper limit) of application- defined warning code range

#### Remarks

Original view error codes are between 100 and 999. Extended view error codes are between 1000 and 8999. Application-definded error codes are between 9000 (VC\_APPLICATION\_DEFINED\_BASE) and 9999 (VC\_APPLICATION\_DEFINED\_END). The normal view warning code is -1. (The old view warning code is 10.) Application-defined view warning codes are between -1999

(VC\_WARNING\_APPLICATION\_DEFINED\_BASE) and -1000 (VC\_WARNING\_APPLICATION\_DEFINED\_END). All return codes that are less than 0 are treated as warnings.

Constants that have "OLD" in them are legacy return codes. Views should avoid returning them and use the newer return codes instead.

Individual views may generate return codes that aren't part of this enumeration. (For example, a view may generate an application-defined code that is 9010. You can get that code by testing for VC\_APPLICATION\_DEFINED\_BASE + 10). A view may also generate a return code that matches a constant's value, but its meaning is different than the constant's meaning.

## tagViewSecurityEnum

Indicates the functionality the user is permitted to access on the view. A security permission can be a combination of multiple defined values.

Name	Value	Description
VIEW_SEC_ADD	1	New records can be added.
VIEW_SEC_MODIFY	2	Existing records can be changed.
VIEW_SEC_DELETE	4	Records can be deleted.
VIEW_SEC_INQUIRE	8	Records can be read.
VIEW_SEC_POST	16	Records can be posted.

### AccpacActiveApp.Item

Retrieves an <u>AccpacApplication</u> object specified by its 0-based index in the collection, or by its two-letter application ID.

Function Item(Index As Variant) As AccpacApplication

#### **Parameters**

Index

[in] the 0-based index of the item in the collection, or the two-letter application ID

#### **Return Value**

Returns the AccpacApplication object corresponding to the supplied index.

#### Remarks

Because the Item method is the default method for this collection, you do not have to specify "Item." For example, the following syntax forms are interchangeable:

activeApp.Item(0)
activeApp(0)
activeApp.Item("name")
activeApp("name")

### **Example**

The following example displays the version of Administrative Services in a message box:

Dim Session As AccpacCOMAPI.AccpacSession
Set Session = CreateObject("ACCPAC.Session")
Session.Open "ADMIN", "ADMIN", "SAMLTD", Date, 0, ""
Dim mDBLinkSysRW As AccpacCOMAPI.AccpacDBLink
Set mDBLinkSysRW = Session.OpenDBLink(DBLINK\_SYSTEM, DBLINK\_FLAG\_READWRITE)
MsgBox mDBLinkSysRW.GetActiveApp.Item("AS").PgmVer

## **AccpacActiveApp.Count**

Returns the number of activated applications (<u>AccpacApplication</u> objects) in the current company database.

Property Count As Long (read-only)

## **AccpacActiveApp Object**

The AccpacActiveApp represents a collection of <u>AccpacApplication</u> objects that describe the activated applications of an ACCPAC company database.

#### **Remarks**

An object of this class cannot be created directly by an application. It should be obtained from an <u>AccpacDBLink</u> object's <u>GetActiveApp</u> property.

# **AccpacApplication.DataLevel**

Returns the application's data level.

Property DataLevel As Integer (read-only)

### **Remarks**

An application's data level must be incremented whenever a new version of an application requires that its data be converted. The data level may be the same for different versions of an application if the version change doesn't require data conversion.

## AccpacApplication.IsInstalled

Returns whether the application module's program files are installed on the machine.

Property IsInstalled As Boolean (read-only)

#### **Remarks**

When an application is activated on a company, the application's tables are created in the company database. Opening a company with a particular application activated does not mean that the application's program files are actually installed on the machine. This property indicates whether the program files are present on the machine.

## AccpacApplication.PgmID

Returns the two-letter program ID of an add-in, or base application (if the application is not an add-in).

Property PgmID As String (read-only)

#### Remarks

For an add-in, the PgmID property returns the program ID of the add-in, and the <u>Selector</u> property returns the program ID of the base application.

The program ID is also referred to as the application ID (or app ID). For example, "AS" is the program ID for Administrative Services.

## AccpacApplication.PgmVer

Returns the three-character program version of the current application.

Property PgmVer As String (read-only)

#### **Remarks**

For example, the program version 51A includes the major version number (5), the minor version number (1), and a revision letter (A).

## **AccpacApplication.Selector**

Returns the program ID of the base application.

Property Selector As String (read-only)

#### **Remarks**

For an add-in, the Selector property returns the program ID of the base application, while the <u>PgmID</u> property returns the program ID of the add-in.

The program ID is also referred to as the application ID (or app ID). For example, "AS" is the program ID for Administrative Services.

## **AccpacApplication.Sequence**

Returns the add-in sequence number.

Property Sequence As String (read-only)

### **Remarks**

For the base application, the sequence number must be "00".

## **AccpacApplication Object**

The AccpacApplication object provides details about an activated application for a company.

### **Remarks**

This object cannot be created directly by an application, but must be obtained from an <a href="AccpacActiveApp">AccpacActiveApp</a> collection.

Returns the first line of the company's address.

Property Address1 As String (read-only)

Returns the second line of the company's address.

Property Address2 As String (read-only)

Returns the third line of the company's address.

Property Address3 As String (read-only)

Returns the fourth line of the company's address.

Property Address4 As String (read-only)

# AccpacCompany.BranchCode

Returns the company's branch code.

Property BranchCode As String (read-only)

# **AccpacCompany.City**

Returns the city where the company is located.

Property City As String (read-only)

# **AccpacCompany.Contact**

Returns the name of the company's contact person.

Property Contact As String (read-only)

# **AccpacCompany.Country**

Returns the name of the country where the company is located.

Property Country As String (read-only)

# AccpacCompany.CountryCode

Returns the company's three-character country code.

Property CountryCode As String (read-only)

# **AccpacCompany.EuroCurrency**

Returns whether the company's home (functional) currency is a Euro currency.

Property EuroCurrency As Boolean (read-only)

#### Remarks

The flag indicating the company's use of a Euro home currency is set (only once) in the Company Profile UI of ACCPAC's Common Services.

# AccpacCompany.Fax

Returns the company's fax number.

Property Fax As String (read-only)

## AccpacCompany.FiscalPeriods

Returns the number of fiscal periods (either 12 or 13) in the company's fiscal year.

Property FiscalPeriods As Integer (read-only)

## AccpacCompany.FourPeriodQuarter

If the number of <u>fiscal periods</u> is 13, then this property returns the fiscal quarter (1, 2, 3, or 4) that contains four periods. The other three fiscal quarters each contain three periods.

Property FourPeriodQuarter As Integer (read-only)

# AccpacCompany.HandleInactiveGLAccou

Returns how inactive General Ledger accounts should be handled.

Property HandleInactiveGLAccounts As tagCompanyHandleInactiveGLAccounts (read-only)

## AccpacCompany.HandleLockedFscPerioc

Returns how locked fiscal periods should be handled.

Property HandleLockedFscPeriods As tagCompanyHandleLockedFscPeriods (read-only)

## AccpacCompany.HandleNonexistantGLA

Returns how non-existent General Ledger accounts should be handled.

Property HandleNonexistantGLAccounts As <a href="mailto:tagCompanyHandleNonexistantGLAccounts">tagCompanyHandleNonexistantGLAccounts</a> (read-only)

## **AccpacCompany.HomeCurrency**

Returns the company's home currency code.

Property HomeCurrency As String (read-only)

### **Remarks**

The home currency is also referred to as the functional currency.

# AccpacCompany.LocationCode

Returns the location code for a company located in a country that requires Value Added Tax (VAT) reporting.

Property LocationCode As String (read-only)

# AccpacCompany.LocationType

Returns the six-character location type code for a company located in a country that requires Value Added Tax (VAT) reporting.

Property LocationType As String (read-only)

# **AccpacCompany.Multicurrency**

Returns whether the company is set up to use multicurrency.

Property Multicurrency As Boolean (read-only)

# AccpacCompany.Name

Returns the company's name.

Property Name As String (read-only)

## AccpacCompany.OrgID

Returns the organization ID (Database ID) of the company as set up in Database Setup.

Property OrgID As String (read-only)

#### Remarks

The company's organization ID cannot be edited in the Company Profile UI of ACCPAC's Common Services. It must be set up in Database Setup.

# AccpacCompany.Phone

Returns the company's telephone number.

Property Phone As String (read-only)

## AccpacCompany.PhoneFormat

Returns whether the company's phone and fax numbers are formatted for display.

Property PhoneFormat As Boolean (read-only)

#### Remarks

Returns TRUE is the telephone numbers are formatted; otherwise, returns FALSE.

# AccpacCompany.PhoneMask

Returns the display mask used to format the company telephone and fax numbers.

Property PhoneMask As String (read-only)

# AccpacCompany.PostCode

Returns the company's postal or zip code.

Property PostCode As String (read-only)

## AccpacCompany.RateType

Returns the company's default currency rate type code (two-character code).

Property RateType As String (read-only)

#### Remarks

The currency rate type is used when converting a source currency amount to its equivalent functional/home currency amount.

### **AccpacCompany.ReportingCurrency**

Returns the currency code of the currency to use in the company's financial reports.

Property ReportingCurrency As String (read-only)

#### **Remarks**

If the **EuroCurrency** property is FALSE, ReportingCurrency is not used.

If the <u>EuroCurrency</u> property is TRUE, ReportingCurrency defines the currency for which fiscal results are to be reported. This may be the Euro (EUR) or one of the Euro's block member currencies.

### AccpacCompany.SessionWarnDays

Returns the number of days that a date can deviate from the session date without generating a warning message.

Property SessionWarnDays As Integer (read-only)

#### **Remarks**

The warning date range is the maximum number of days that a transaction date, such as an invoice date, can deviate from the <u>session</u> <u>date</u> without generating a warning message.

# **AccpacCompany.State**

Returns the province or state where the company is located.

Property State As String (read-only)

### **AccpacCompany Object**

The AccpacCompany object provides company profile information (from the Company Profile UI of ACCPAC's Common Services) for the company you specified when you opened the session (refer to the Open method of the AccpacSession object).

#### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from an <u>AccpacDBLink</u> object's <u>GetCompany</u> method, which provides information about the company to which the database link is connected.

### AccpacCurrency.IsBlockCombinationWith

Returns whether the current AccpacCurrency object's currency belongs to the same currency block as the specified currency code, on or before the specified date.

```
Function IsBlockCombinationWith(
CurCode As String,
Date As Date,
BlockDateMatch As tagBlockDateMatchEnum) As Boolean
```

#### **Parameters**

```
CurCode
[in] the currency code

Date
[in] the date

BlockDateMatch
[in] how date matching should be done
```

#### **Return Value**

Returns whether the current AccpacCurrency object's currency belongs to the same currency block as *CurCode*, on or before *Date*.

### AccpacCurrency.IsBlockMaster

Returns whether the current AccpacCurrency object's currency is a block master currency on or before the given date.

```
Function IsBlockMaster(
Date As Date,
BlockDateMatch As tagBlockDateMatchEnum) As Boolean
```

### **Parameters**

```
Date
[in] the date

BlockDateMatch
[in] how date matching should be done
```

#### **Return Value**

Returns whether the current AccpacCurrency object's currency is a block master currency on or before *Date*.

### AccpacCurrency.IsBlockMember

Returns whether the currency is a member of a block currency on or before the given date. If it's TRUE, the currency rate information with the block master is also returned.

Function IsBlockMember(
Date As Date,
BlockDateMatch As tagBlockDateMatchEnum,
[CurrencyRate As AccpacCurrencyRate]) As Boolean

#### **Parameters**

Date
[in] the date

BlockDateMatch

[in] how date matching should be done

CurrencyRate [out, optional] currency rate information with block master

#### **Return Value**

Returns whether the currency is a member of a block currency on or before *Date*.

## **AccpacCurrency.Decimals**

Returns the number of decimals places (0, 1, 2, or 3) to use for the currency.

Property Decimals As Integer (read-only)

## AccpacCurrency.DecimalSep

Returns the decimal group separator.

Property DecimalSep As String (read-only)

#### Remarks

Returns the character that separates whole numbers from decimals. The separator can be "." (period), "," (comma), or " " (blank).

## **AccpacCurrency.Description**

Returns the name of the currency (the description associated with the currency code).

Property Description As String (read-only)

## **AccpacCurrency.NegativeDisplay**

Returns the display format to be used for negative values.

Property NegativeDisplay As <u>tagNegativeDisplayEnum</u> (*read-only*)

## **AccpacCurrency.Symbol**

Returns the currency code symbol (the standard abbreviation or symbol used for the currency).

Property Symbol As String (read-only)

## AccpacCurrency.SymbolDisplay

Returns the currency symbol's display position and format.

Property SymbolDisplay As <u>tagCurrencySymbolDisplayEnum</u> (*read-only*)

#### **Remarks**

The currency symbol can be displayed before or after the amount, with or without a space between it and the amount.

## AccpacCurrency.ThousandSep

Returns the thousands group separator (character that separates thousands in large numbers) to use for the currency.

Property ThousandSep As Integer (read-only)

#### Remarks

The separator can be "." (period), "," (comma), or " " (blank).

### **AccpacCurrency Object**

The AccpacCurrency object provides currency information for the currency code that you pass to the <u>GetCurrency</u> method of the <u>AccpacSession</u> object.

The currency code is set up in the Currency/Currency Codes UI of ACCPAC's Common Services.

#### Remarks

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetCurrency</u> method.

### AccpacCurrencyRate.DateMatch

Returns the date matching method to use when determining the source currency exchange rate.

Property DateMatch As <a href="mailto:tagDateMatchEnum">tagDateMatchEnum</a> (read-only)

#### Remarks

Date matching is only used in multicurrency databases when a transaction requires currency conversion. Date matching enables ACCPAC to determine the source currency exchange rate, when no rate exists for the date of the transaction or revaluation.

## **AccpacCurrencyRate.HomeCurrency**

Returns the home (functional) currency code of the exchange rate.

Property HomeCurrency As String (read-only)

# AccpacCurrencyRate.Rate

Returns the exchange rate.

Property Rate As Double (read-only)

# AccpacCurrencyRate.RateDate

Returns the effective date for the exchange rate.

Property RateDate As Date (read-only)

# AccpacCurrencyRate.RateOperator

Returns the operator to be used for currency conversion.

Property RateOperator As <a href="mailto:tagRateOperatorEnum">tagRateOperatorEnum</a> (read-only)

## AccpacCurrencyRate.RateType

Returns the currency rate type code used to determine the exchange rate.

Property Ratetype As String (*read-only*)

## AccpacCurrencyRate.SourceCurrency

Returns the source currency code (the transaction currency that will be converted to the functional currency equivalent) of the currency rate.

Property SourceCurrency As String (read-only)

# AccpacCurrencyRate.Spread

Returns the allowed difference that the rate entered for a transaction can vary from the actual exchange rate entered in the currency table.

Property Spread As Double (read-only)

## **AccpacCurrencyRate Object**

The AccpacCurrencyRate provides currency rate information as set up in the Currency/Currency Rates UI of ACCPAC's Common Services.

## Remark

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetCurrencyRate</u>, <u>GetCurrencyRateComposite</u>, and <u>GetCurrencyRateFloating</u> methods.

# AccpacCurrencyTable.DateMatch

Returns the date matching method used to determine exchange rates.

Property DateMatch As <a href="mailto:tagDateMatchEnum">tagDateMatchEnum</a> (read-only)

# AccpacCurrencyTable.Description

Returns the currency table's description.

Property Description As String (read-only)

# AccpacCurrencyTable.RateOperator

Returns the rate operator to be used for currency conversion.

Property RateOperator As <a href="mailto:tagRateOperatorEnum">tagRateOperatorEnum</a> (read-only)

# AccpacCurrencyTable.SourceOfRates

Returns the source from which the exchange rates are quoted.

Property SourceOfRates As String (read-only)

## **Remarks**

Examples of "source" are newspaper and website.

## **AccpacCurrencyTable Object**

The AccpacCurrencyTable object represents a currency table set up in the Currency/Currency Rates UI of ACCPAC's Common Services. It provides information about how to use exchange rates for the currency codes stored in the system.

## **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetCurrencyTable</u> method.

# AccpacDBLink.Close

Closes the database link.

Sub Close()

## **Remarks**

A closed AccpacDBLink object cannot be reopened. To initiate another database link, a new AccpacDBLink object must be opened from an <u>AccpacSession</u> object.

# AccpacDBLink.CreateViewTables

Instructs the specified view to create the tables that it accesses on the current database link.

Sub CreateViewTables(ViewID As String)

## **Parameters**

ViewID [in] the Roto ID of the view

### **Remarks**

This method loads the specified view on the current database link and calls viewCreate() to create the tables that the view needs.

## AccpacDBLink.DropViewTables

Instructs the specified view to drop (delete) its tables (tables created by <u>CreateViewTables</u>) on the current database connection.

Sub DropViewTables(ViewID As String)

### **Parameters**

ViewID [in] the Roto ID of the view

### **Remarks**

This method loads the specified view and calls viewDrop() to drop the tables that it accesses.

# AccpacDBLink.GetActiveApp

Returns an <u>AccpacActiveApp</u> object that provides access to details of all the active applications of the current company database.

Function GetActiveApp() As <u>AccpacActiveApp</u>

### **Return Value**

Returns an <u>AccpacActiveApp</u> object, which stores a collection of active applications for the current database.

# **AccpacDBLink.GetCompany**

Returns an <u>AccpacCompany</u> object that contains the company profile settings.

Function GetCompany() As <a href="https://www.nction.com/AccpacCompany">AccpacCompany</a>

## **Return Value**

Returns the AccpacCompany object.

## AccpacDBLink.GetFiscalCalendar

Returns an <u>AccpacFiscalCalendar</u> object that provides access to the fiscal calendar of the current company.

Function GetFiscalCalendar() As <a href="AccpacFiscalCalendar">AccpacFiscalCalendar</a>

## **Return Value**

Returns the AccpacFiscalCalendar object of the current company.

## AccpacDBLink.GetProcessServerSetup

Creates and returns an AccpacProcessServerSetup object for the view identified by the supplied ViewID. If a view has been configured in the Process Server Administrator to be run via Process Server, then the view must be opened using <a href="OpenViewExt">OpenViewExt</a> (pass in the configured AccpacProcessServerSetup object as the second parameter).

Function GetProcessServerSetup(ViewID As String) As AccpacProcessServerSetup

#### **Parameters**

ViewID [in] target view's Roto ID

### **Return Value**

Returns the AccpacProcessServerSetup object for the specified view.

## Remarks

The AccpacProcessServerSetup object is a hidden member of the <u>ACCPAC COM API</u> library.

## AccpacDBLink.OpenView

Opens a view on the current database link.

```
Function OpenView(
ViewID As String,
ppView As <u>AccpacView</u>) As Long
```

## **Parameters**

```
ViewID
[in] the Roto ID of the view to open

ppView
[out] returns the AccpacView object created
```

#### **Return Value**

Returns a status code of the view open operation. If the view is opened successfully, the return value is 0.

## **Remarks**

If a view has been configured to be run via Process Server, OpenView will fail and return VC\_OPEN\_FAILED\_PS. In this case, an application should call <u>GetProcessServerSetup</u>, configure the returned object, and then call <u>OpenViewExt</u>, passing in the configured AccpacProcessServerSetup object.

## AccpacDBLink.OpenView2

Opens a view on the current database link.

```
Function OpenView2(
ViewID As String,

[pExtErrInfo] As Variant) As <u>AccpacView</u>
```

## **Parameters**

```
ViewID
[in] the Roto ID of the view to open

pExtErrInfo
[optional, in]
```

### **Return Value**

Returns an <u>AccpacView</u> object.

## **Remarks**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <a href="OpenView">OpenView</a> function.

## AccpacDBLink.OpenViewExt

Opens a view on the current database link, using the Process Server settings specified in the AccpacProcessServerSetup object.

```
Function OpenViewExt(
ViewID As String,
pVal As AccpacProcessServerSetup,
ppView As AccpacView) As Long
```

#### **Parameters**

```
ViewID
  [in] the Roto ID of the view to open
pVal
  [in] an AccpacProcessServerSetup object configured with the desired settings
ppView
  [out] returns the <a href="AccpacView">AccpacView</a> object created
```

#### **Return Value**

Returns a status code of the view open operation. If the view is successfully opened, the function returns 0. If the Process Server does not respond, the function fails with return code VC\_OPEN\_FAILED\_PSCONNECT.

The AccpacProcessServerSetup object is a hidden member of the <u>ACCPAC COM API</u> library.

## AccpacDBLink.OpenViewInstance

Opens a view on the current database link, with the optional arguments giving enhanced control over the view's behavior.

```
Function OpenViewInstance(
ViewID As String,

ppView As AccpacView,

[Readonly As Boolean = False],

[Unvalidated As Boolean = False],

[Unrevisioned As Boolean = False],

[Nonheritable As Boolean = False],

[Prefetch As Long],

[RawPut As Boolean = False],

[Noncascading As Boolean = False],

[extra] As Variant,

[ProcessServer As AccpacProcessServerSetup]) As Long
```

### **Parameters**

```
ViewID
[in] the Roto ID of the view to open

ppView
[out] returns the AccpacView object created

Readonly
[in, optional] if set to TRUE, opens the view in Read-only mode

Unvalidated
[in, optional] if set to TRUE, suppresses validation on the view

Unrevisioned
[in, optional] if set to TRUE, revisioning is disabled for the view

Nonheritable
```

[in, optional] if set to TRUE, composite views will not inherit the flags with which this view was opened

#### Prefetch

[in, optional] the number of records to pre-fetch when the view instance is read-only

#### RawPut

[in, optional] if set to TRUE, data will not be processed upon a field Put

## Noncascading

[in, optional] if set to TRUE, suppresses propagation of foreign key values to composite views

#### extra

[in, optional] application-specific data to be passed to the view when it is opened (an unstructured block of data given to the view - the data type and format should be understood by both the caller and the view)

#### **ProcessServer**

[in, optional] an AccpacProcessServerSetup object configured with the desired settings; pass Null if Process Server should not be used with the view

### **Return Value**

Returns a status code of the view open operation. If the view is opened successfully, the return value is 0.

## Remarks

The AccpacProcessServerSetup object is a hidden member of the ACCPAC COM API library.

## AccpacDBLink.OpenViewInstance2

Opens a view on the current database link, with the optional arguments giving enhanced control over the view's behavior.

```
Function OpenViewInstance(
ViewID As String,

[ProcessServer As AccpacProcessServerSetup],

[pExtErrInfo] As Variant,

[Readonly As Boolean = False],

[Unvalidated As Boolean = False],

[Unrevisioned As Boolean = False],

[Nonheritable As Boolean = False],

[Prefetch As Long],

[RawPut As Boolean = False],

[Noncascading As Boolean = False],

[extra] As Variant) As AccpacView
```

### **Parameters**

#### **ViewID**

[in] the Roto ID of the view to open

#### **ProcessServer**

[in, optional] an AccpacProcessServerSetup object configured with the desired settings; pass Null if Process Server should not be used with the view

## pExtErrInfo

[in, optional]

## Readonly

[in, optional] if set to TRUE, opens the view in Read-only mode

#### Unvalidated

[in, optional] if set to TRUE, suppresses validation on the view

#### Unrevisioned

[in, optional] if set to TRUE, revisioning is disabled for the view

#### Nonheritable

[in, optional] if set to TRUE, composite views will not inherit the flags with which this view was opened

#### Prefetch

[in, optional] the number of records to pre-fetch when the view instance is read-only

#### RawPut

[in, optional] if set to TRUE, data is will not be processed upon a field put

### Noncascading

[in, optional] if set to TRUE, suppresses propagation of foreign key values to composite views

#### extra

[in, optional] application-specific data to be passed to the view when it is opened (an unstructured block of data given to the view - the data type and format should be understood by both the caller and the view)

### **Return Value**

Returns an **AccpacView** object.

## Remarks

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <a href="OpenViewInstance">OpenViewInstance</a> function.

## AccpacDBLink.ParamGet

Gets the values of the specified field IDs from the first record of the specified view.

Function ParamGet(
ViewID As String,
FieldIDs as Variant) As Variant

### **Parameters**

**ViewID** 

[in] the Roto ID of the view whose field values (for the first record) are to be returned

**FieldIDs** 

[in] an array of field IDs (array of Longs)

### **Return Value**

Returns an array of variants containing the values of the requested fields in the view's first record. The values are returned in the same order as requested field IDs as they appear in the FieldIDs array.

## Remarks

This method is applicable only to views that are designed to store parameters of an application. These views typically have one and only one record. Internally, this method opens the specified view, fetches the first record, and returns the values of the specified field IDs.

In a networked environment, this method is more efficient than manually opening a view, and then fetching and obtaining field values, since this method only takes one network call and all processing is done on the server.

Unlike the BlkGet method of the DataSource control, this method will access the view's values rather than cached values.

# AccpacDBLink.SecCheck

Performs a security check for the current user on the specified security resource ID.

Function SecCheck(rscID As String) As Boolean

### **Parameters**

rscID

[in] the security resource ID on which to perform the security check

#### **Return Value**

Returns whether the current user has access rights to the specified security resource.

## AccpacDBLink.TransactionBegin

Begins a transaction on the current database connection.

Function TransactionBegin(pTransLevel As Long) As Long

### **Parameters**

pTransLevel [out] returns the transaction level after the transaction begins

## **Return Value**

Returns a status code from the database layer. If the operation was successful, the function returns 0. If the function failed, it returns an error code as defined in DBS.H.

## AccpacDBLink.TransactionBegin2

Begins a transaction on the current database connection.

Function TransactionBegin2(pTransLevel As Variant) As Long

### **Parameters**

*pTransLevel*[out] returns the transaction level after the transaction begins

## **Return Value**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>TransactionBegin</u> function.

## **AccpacDBLink.TransactionCommit**

Commits the current transaction on the database connection.

Function TransactionCommit(pTransLevel As Long) As Long

### **Parameters**

pTransLevel [out] returns the transaction level after the transaction is committed

## **Return Value**

Returns a status code from the database layer. If the operation was successful, the function returns 0. If the function failed, it returns an error code as defined in DBS.H.

## AccpacDBLink.TransactionCommit2

Commits the current transaction on the database link.

Function TransactionCommit2(pTransLevel As Variant) As Long

### **Parameters**

pTransLevel [out] returns the transaction level after the transaction is committed

## **Return Value**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>TransactionCommit</u> function.

## AccpacDBLink.TransactionGetLevel

Returns the current transaction level on the database connection.

Function TransactionGetLevel(pTransLevel As Long) As Long

### **Parameters**

pTransLevel [out] returns the current transaction level

## **Return Value**

Returns a status code from the database layer. If the operation was successful, the function returns 0. If the function failed, it returns an error code as defined in DBS.H.

## AccpacDBLink.TransactionGetLevel2

Returns the current transaction level on the database connection.

Function TransactionGetLevel2(pTransLevel As Variant) As Long

### **Parameters**

pTransLevel [out] returns the current transaction level

## **Return Value**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>TransactionGetLevel</u> function.

## AccpacDBLink.TransactionRollback

Rolls back the current transaction on the database connection.

Function TransactionRollback(pTransLevel As Long) As Long

### **Parameters**

pTransLevel
[out] returns the transaction level after the current transaction is rolled back

#### **Return Value**

Returns a status code from the database layer. If the operation was successful, the function returns 0. If the function failed, it returns an error code as defined in DBS.H.

## AccpacDBLink.TransactionRollback2

Rolls back the current transaction on the database connection.

Function TransactionRollback2(pTransLevel As Variant) As Long

### **Parameters**

pTransLevel
[out] returns the transaction level after the current transaction is rolled back

## **Return Value**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>TransactionRollback</u> function.

# AccpacDBLink.Flags

Returns the read/write access mode of the database link.

Property Flags As <u>tagDBLinkFlagsEnum</u> (read-only)

## **Remarks**

The access mode of the link is set when the AccpacDBLink object is created from the <a href="OpenDBLink">OpenDBLink</a> method of an <a href="AccpacSession">AccpacSession</a> object.

# **AccpacDBLink.Session**

Returns the <u>AccpacSession</u> object that created the current AccpacDBLink object.

Property Session As <u>AccpacSession</u> (*read-only*)

# AccpacDBLink.Type

Returns the type of the database link.

Property Type As <a href="mailto:tagDBLinkTypeEnum">tagDBLinkTypeEnum</a> (read-only)

## **Remarks**

The database link type (company or system) is set when the AccpacDBLink object is created by calling the <a href="OpenDBLink">OpenDBLink</a> method of an <a href="AccpacSession">AccpacSession</a> object.

## **AccpacDBLink Object**

The AccpacDBLink object represents a connection (link) to an ACCPAC database.

There are two types of ACCPAC databases:

- Company: A company database stores application data (covers most Common Services functions and all other applications). Each company database must be linked to a system database.
- System: A system database stores the data common to multiple company databases, such as currency information, user accounts, and authorization information. A system database can be associated with multiple company databases.

### Remarks

An object of this class cannot be created directly by applications. Instead, it should be obtained from the <u>AccpacSession</u> object's <u>OpenDBLink</u> method.

Multiple AccpacDBLink objects created on the same database represent multiple connections opened to the database.

# **AccpacErrors.Clear**

Clears all errors in the collection.

Sub Clear()

### **Remarks**

Clears the error stack.

## **Example**

The following example clears the Errors collection after the program displays each error.

```
Dim Errors As AccpacErrors
Dim nCnt As Long
Set Errors = Session.Errors
nCnt = Errors.Count
If nCnt = 0 Then
    MsgBox Err.Description
Else
    Dim idx As Long
    For idx = 0 To (nCnt - 1)
        MsgBox Errors(idx)
    Next idx
    Errors.Clear
End If
```

# AccpacErrors.GenerateErrorFile

Generates a temporary file to store the errors in the collection.

Function GenerateErrorFile() As String

#### **Return Value**

Returns the full path, including the file name, of the generated CSV (comma delimited) file.

#### Remarks

When ACCPAC is running remotely, the temporary file is generated on the server, so the returned path is a path on the server machine (a DOS path that is "local" to the server, as opposed to a URL).

This method is useful for applications wanting to generate a report of all errors stored for the session.

# **AccpacErrors.Get**

Retrieves an error message based on its 0-based index in the collection. This method also returns all of the related information for the error message.

```
Sub Get(
Index As Long,
pMsg As String,
pPriority As tagErrorPriority,
pSource As String,
pErrCode As String,
pHelpFile As String,
pHelpID As Long)
```

#### **Parameters**

```
Index
[in] the 0-based index of the error message in the list

pMsg
[out] the error message

pPriority
[out] the error priority

pSource
[out] the source of error

pErrCode
[out] the error code

pHelpFile
[out] the help file in which help messages can be found for this error

pHelpID
```

# [out] the Context ID of the help message

## Remarks

# AccpacErrors.Get2

Retrieves an error message based on its 0-based index in the collection. This method also returns all of the related information for the error message.

```
Sub Get(
Index As Long,
pMsg As Variant,
pPriority As Variant,
pSource As Variant,
pErrCode As Variant,
pHelpFile As Variant,
pHelpID As Long)
```

#### **Parameters**

```
Index
[in] the 0-based index of the error message in the list

pMsg
[out] the error message

pPriority
[out] error priority

pSource
[out] the source of error

pErrCode
[out] the error code

pHelpFile
[out] the help file in which help messages can be found for this error

pHelpID
```

## [out] the Context ID of the help message

## Remarks

Script languages should use this variation of the <u>Get</u> function.

# **AccpacErrors.Item**

Returns an error message based on its 0-based index in the collection.

Function Item(Index As Long) As String (default)

#### **Parameters**

Item

[in] the 0-based index of the error message in the collection

### **Return Value**

Returns the error message at he specified index. To obtain other information about the error, such as the source or error code, use <u>Get</u>.

### **Remarks**

Because the Item method is the default method of this collection, you do not have to specify "Item." For example, the following syntax forms are interchangeable:

Errors.Item(0) Errors(0)

### **Example**

Refer to the example in the <u>Clear</u> method.

# **AccpacErrors.Put**

Adds an error message to the collection.

```
Sub Put(
Msg As String,
Priority As tagErrorPriority,
[Params As Variant],
[Source As String],
[ErrCode As String],
[HelpFile As String],
[HelpContextID As Long])
```

#### **Parameters**

```
Msg
[in] the error message

Priority
[in] the error priority
```

#### **Params**

[in, optional] an array of parameters for the error message if the error message contains string-replacement tokens, or Null if replacement tokens are not used in the error message. The tokens in the error message will be replaced by the specified parameters.

#### Source

[in, optional] the source of the error

#### **ErrCode**

[in, optional] the error code

### HelpFile

[in, optional] the help file in which help messages can be found for

this error

HelpContextID [in, optional] the Context ID of the help message

### **Remarks**

The error message could be plain text, or a string with replacement tokens. Replacement tokens take the form %n, where n is a number starting from 1. When replacement tokens are present in the error message, they are replaced by the value in the corresponding position in the *Params* array.

Use <u>PutRscMsg</u> if the error message comes from an application's serverside language resource file.

# AccpacErrors.PutRscMsg

Adds an error message to the collection from an application's server-side language resource file.

```
Sub PutRscMsg(
AppID As String,
rscID As Long,
Priority As tagErrorPriority,
[Params As Variant],
[Source As String],
[ErrCode As String],
[HelpFile As String],
[HelpContextID As Long])
```

#### **Parameters**

#### **AppID**

[in] the 2-character application ID of the language resource file

#### rscID

[in] the resource ID of the error message defined in the language resource file

### Priority

[in] the error priority

#### **Params**

[in, optional] an array of parameters for the error message if the error message contains string-replacement tokens, or Null if replacement tokens are not used in the error message. The tokens in the error message will be replaced by the specified parameters.

#### Source

[in, optional] the source of the error

#### **ErrCode**

[in, optional] the error code

#### HelpFile

[in, optional] the help file in which help messages can be found for this error

### **HelpContextID**

[in, optional] the Context ID of the help message

#### Remarks

The error message is retrieved from the application's server-side string resource file identified by the *ApplD* and *rscID*. The error message could be plain text, or a string with replacement tokens. Replacement tokens take the form %n, where n is a number starting from 1. When replacement tokens are present in the error message, they are replaced by the value in the corresponding position in the *Params* array.

PutRscMsg can be used to display error messages whose strings come from other applications. For example, a Bank Services UI can use PutRscMsg to display an error message whose string comes from the Common Services server-side language resource file.

# **AccpacErrors.Count**

Returns the number of errors contained in the collection.

Property Count As Long (read-only)

## Remarks

# **AccpacErrors Object**

The AccpacErrors object provides access to error and information messages, and to details generated during the current session.

### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>Errors</u> property.

This object stores all the errors raised within the current session. This includes errors raised from:

- calling methods/properties of the Session object
- calling methods/properties of all objects created directly or indirectly from the Session object
- all views created within the session

# AccpacFiscalCalendar.DatesFromPeriod

Calculates the period start and end dates, given a period number, type, and length, and a base date.

Function DatesFromPeriod(
Period As Integer,
PeriodType As tagPeriodType,
PeriodLength As Integer,
BaseDate As Date,
StartDate As Date,
EndDate As Date) As Boolean

#### **Parameters**

[in] the period number

PeriodType
[in] the period type

PeriodLength
[in] the period length

BaseDate
[in] the starting date of the first period

StartDate
[out] returns the period's start date

EndDate
[out] returns the period's end date

### **Return Value**

Returns whether the start and end dates were successfully determined.

## **Remarks**

Use a combination of period type and length to denote additional period types.

## **Examples:**

- Bimonthly: PeriodType = PERTYPE\_MONTHLY, PeriodLength = 2
- Biweekly (with Sunday = day 1):
   PeriodType = PERTYPE\_WEEKLY, PeriodLength = 14

# AccpacFiscalCalendar.DatesFromPeriod2

Calculates the period start and end dates, given a period number, type, and length, and a base date.

Function DatesFromPeriod2(
Period As Integer,
PeriodType As tagPeriodType,
PeriodLength As Integer,
BaseDate As Date,
StartDate,
EndDate) As Boolean

#### **Parameters**

Period
[in] the period number

PeriodType
[in] the period type

PeriodLength
[in] the period length

BaseDate
[in] the starting date of the first period

StartDate
[out] returns the period's start date

EndDate
[out] returns the period's end date

#### **Return Value**

Returns whether the start and end dates were successfully determined.

## Remarks

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <a href="DatesFromPeriod">DatesFromPeriod</a> function.

# AccpacFiscalCalendar.DateToPeriod

Calculates the period number, given a date, period type and length, and a base date.

Function DateToPeriod(
Date As Date,
PeriodType As tagPeriodType,
PeriodLength As Integer,
BaseDate As Date,
Period As Integer) As Boolean

#### **Parameters**

Date

[in] the date

PeriodType

[in] the period type

PeriodLength

[in] the period length

BaseDate

[in] the starting date of the first period

Period

[out] returns the period number that contains the specified date

## **Return Value**

Returns whether the period number was successfully determined.

### Remarks

Use a combination of period type and length to denote additional period types.

## **Examples:**

- Bimonthly: PeriodType = PERTYPE\_MONTHLY, PeriodLength = 2
- Biweekly (with Sunday = day 1):
   PeriodType = PERTYPE\_WEEKLY, PeriodLength = 14

# AccpacFiscalCalendar.DateToPeriod2

Calculates the period number, given a date, period type and length, and a base date.

Function DateToPeriod2(
Date As Date,
PeriodType As tagPeriodType,
PeriodLength As Integer,
BaseDate As Date,
Period) As Boolean

#### **Parameters**

Date

[in] the date

PeriodType

[in] the period type

PeriodLength

[in] the period length

BaseDate

[in] the starting date of the first period

Period

[out] returns the period number that contains the specified date

## **Return Value**

Returns whether the period number was successfully determined.

### Remarks

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>DateToPeriod</u> function.

# AccpacFiscalCalendarGetFirstYear

Retrieves information on the first fiscal year set up in the company's fiscal calendar.

Function GetFirstYear(
[Year As String],
[FiscalPeriods As Integer],
[Qtr4Period As Integer],
[Active As Boolean]) As Boolean

#### **Parameters**

Year

[out, optional] returns the fiscal year

**FiscalPeriods** 

[out, optional] returns the number of periods in the fiscal year

Qtr4Period

[out, optional] returns the quarter that contains 4 periods, if the fiscal year has 13 periods

Active

[out, optional] returns whether the fiscal year is active

### **Return Value**

Returns whether the first fiscal year is found in the fiscal calendar.

# AccpacFiscalCalendar.GetLastYear

Retrieves information of the last fiscal year set up in the company's fiscal calendar.

Function GetLastYear(
[Year As String],
[FiscalPeriods As Integer],
[Qtr4Period As Integer],
[Active As Boolean]) As Boolean

#### Year

[out, optional] returns the fiscal year

#### **FiscalPeriods**

[out, optional] returns the number of periods in the fiscal year

#### **Qtr4Period**

[out, optional] returns the quarter that contains 4 periods, if the fiscal year has 13 periods

#### Active

[out, optional] returns whether the fiscal year is active

#### **Return Value**

Returns whether the last fiscal year is found in the fiscal calendar.

# AccpacFiscalCalendar.GetPeriod

Retrieves the fiscal year and period to which a given date belongs.

```
Function GetPeriod(
Date As Date,
[Period As Integer],
[Year As String],
[PeriodOpen As Boolean]) As Boolean
```

#### **Parameters**

Date

[in] the date

Period

[out, optional] returns the fiscal period that includes the specified date

Year

[out, optional] returns the fiscal year that includes the specified date

PeriodOpen

[out, optional] returns whether the fiscal period is open (not locked)

### **Return Value**

Returns whether the specified date falls in a fiscal period defined in the fiscal calendar.

# AccpacFiscalCalendar.GetPeriodDates

Retrieves the start and end dates of the given fiscal year and period.

Function GetPeriodDates(
Year As String,
Period As Integer,
[StartDate As Date],
[EndDate As Date],
[PeriodOpen As Boolean]) As Boolean

#### **Parameters**

```
Year
[in] the fiscal year

Period
[in] the fiscal period

StartDate
[out, optional] returns the fiscal period start date

EndDate
[out, optional] returns the fiscal period end date

PeriodOpen
[out, optional] returns whether the period is open (not locked)
```

### **Return Value**

Returns whether the specified fiscal year and period are defined in the fiscal calendar.

# AccpacFiscalCalendar.GetQuarter

Retrieves the fiscal quarter to which a given fiscal period and year belong.

```
Function GetQuarter(
Year As String,
Period As Integer,
Quarter As Integer) As Boolean
```

#### **Parameters**

```
Year
[in] the fiscal year

Period
[in] the fiscal period

Quarter
[out] returns the quarter the period belongs to
```

### **Return Value**

Returns whether the specified fiscal year and period are defined in the fiscal calendar.

# AccpacFiscalCalendar.GetQuarter2

Retrieves the fiscal quarter to which a given fiscal period and year belong.

```
Function GetQuarter2(
Year As String,
Period As Integer,
Quarter) As Boolean
```

#### **Parameters**

```
Year
[in] the fiscal year

Period
[in] the fiscal period

Quarter
[out] returns the quarter the period belongs to
```

### **Return Value**

Returns whether the specified fiscal year and period are defined in the fiscal calendar.

### **Remarks**

Script languages, such as Javascript, that do not accept "out" parameters, should use this variation of the <u>GetQuarter</u> function.

# AccpacFiscalCalendar.GetQuarterDates

Retrieves the start and end dates of a given fiscal year and quarter.

Function GetQuarterDates(
Year As String,
Quarter As Integer,
[StartDate As Date],
[EndDate As Date]) As Boolean

#### **Parameters**

```
Year
[in] the fiscal year

Quarter
[in] the fiscal quarter

StartDate
[out, optional] returns the quarter's start date

EndDate
[out, optional] returns the quarter's end date
```

#### **Return Value**

Returns whether the specified fiscal year and quarter are defined in the fiscal calendar.

# AccpacFiscalCalendar.GetYear

Retrieves information about the given fiscal year.

Function GetYear(
Year As String,
[FiscalPeriods As Integer],
[Qtr4Period As Integer],
[Active As Boolean]) As Boolean

#### **Parameters**

Year

[in] the fiscal year to locate in the fiscal calendar

*FiscalPeriods* 

[out, optional] returns the number of periods in the fiscal year

Qtr4Period

[out, optional] returns the quarter that contains 4 fiscal periods, if the fiscal year has 13 periods (that is, <u>FiscalPeriods</u> is 13)

Active

[out, optional] returns whether the fiscal year is active

### **Return Value**

Returns whether the specified fiscal year is defined in the fiscal calendar.

# AccpacFiscalCalendar.GetYearDates

Retrieves the start and end dates of a given fiscal year.

Function GetYearDates(
Year As String,
[StartDate As Date],
[EndDate As Date]) As Boolean

### **Parameters**

Year
[in] the fiscal year to locate in the fiscal calendar

StartDate
[out, optional] returns the fiscal year start date

EndDate
[out, optional] returns the fiscal year end date

### **Return Value**

Returns whether the specified fiscal year is defined in the fiscal calendar.

# **AccpacFiscalCalendar Object**

The AccpacFiscalCalendar object provides methods to access the company's fiscal calendar set up in the Fiscal Calendar UI of ACCPAC's Common Services.

### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacDBLink</u> object's <u>GetFiscalCalendar</u> method. The object represents the fiscal calendar of the company to which the database link is connected.

# AccpacMultiuser.LockApp

Locks an application's data as shared (read lock) or exclusive (write lock).

```
Function LockApp(
OrgID As String,
AppID As String,
Exclusive As Boolean) As tagMultiuserStatus
```

#### **Parameters**

### OrgID

[in] the organization (database) ID of the company

### **AppID**

[in] the two-letter application ID of the application whose data is to be locked

#### **Exclusive**

[in] flag indicating whether to place an exclusive (TRUE) or shared (FALSE) lock on the application's data

#### **Return Value**

Returns the status of the call.

### Remarks

If possible, avoid using this method. Locking should be done by views rather than by UIs. Locking (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet. When a view opens, it automatically locks the application's data as "shared", if required, so there's no need to explicitly lock the application's data as "shared."

Call LockApp only when the application's data isn't currently locked by a previous call to LockApp. To upgrade (from shared to exclusive) or downgrade (from exclusive to shared) a program's existing lock on the application's data, use <a href="RegradeApp">RegradeApp</a>. To unlock the application's data, use <a href="UnlockApp">UnlockApp</a>.

# AccpacMultiuser.LockOrg

Locks an organization's database as shared (read lock) or exclusive (write lock).

Function LockOrg(
OrgID As String,
Exclusive As Boolean) As tagMultiuserStatus

#### **Parameters**

OrgID

[in] the organization (database) ID of the database to be locked

**Exclusive** 

[in] flag indicating whether to place an exclusive (TRUE) or shared (FALSE) lock on the organization's database

### **Return Value**

Returns the status of the call.

### Remarks

If possible, avoid using this method. Locking should be done by views rather than by UIs. Locking (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet. When a view opens, it automatically locks the organization's database as "shared", if required, so there's no need to explicitly lock the organization's database as "shared."

Call LockOrg only when the organization's database isn't currently locked by a previous call to LockOrg. To upgrade (from shared to exclusive) or downgrade (from exclusive to shared) a program's existing lock on the organization's database, use <a href="RegradeOrg">RegradeOrg</a>. To unlock the organization's

database, use <u>UnlockOrg</u>.

# AccpacMultiuser.LockRsc

Locks a resource as shared (read lock) or exclusive (write lock).

Function LockRsc(
Resource As String,
Exclusive As Boolean) As <u>tagMultiuserStatus</u>

### **Parameters**

Resource

[in] name of the resource to lock

**Exclusive** 

[in] flag indicating whether to place an exclusive (TRUE) or shared (FALSE) lock on the resource

#### **Return Value**

Returns the status of the call.

#### Remarks

If possible, avoid using this method. Locking should be done by views rather than by UIs. Locking (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet.

Unlike <u>LockApp</u> and <u>LockOrg</u>, to upgrade (from shared to exclusive) or downgrade (from exclusive to shared) an existing lock on the resource, you can call LockRsc with the appropriate value for Exclusive (TRUE to upgrade, FALSE to downgrade) without first having to call <u>UnlockRsc</u>.

To unlock the resource, use **UnlockRsc**.

# AccpacMultiuser.RegradeApp

Upgrades (from shared to exclusive) or downgrades (from exclusive to shared) a lock on an application's data.

Function RegradeApp(
OrgID As String,
AppID As String,
Upgrade As Boolean) As tagMultiuserStatus

#### **Parameters**

#### **OrgID**

[in] organization (database) ID of the company

## **AppID**

[in] two-letter application ID of the application whose lock is being upgraded or downgraded

## Upgrade

[in] flag indicating whether the program's lock on the application's data should be upgraded (TRUE) or downgraded (FALSE)

## **Return Value**

Returns the status of the call.

## Remarks

If possible, avoid using this method. Lock manipulation should be done by views rather than by UIs. Lock manipulation (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet.

Before calling RegradeApp, the program must have already locked the

application's data by calling <u>LockApp</u>. Calls to RegradeApp may be nested. If a program initially has shared access to the application's data, nested calls to RegradeApp to upgrade and then restore the program's lock on the application's data will result in the data being locked exclusively until the outermost level of nested calls restores the lock, returning it to shared access.

# AccpacMultiuser.RegradeOrg

Upgrades (from shared to exclusive) or downgrades (from exclusive to shared) a lock on an organization's database.

Function RegradeOrg(
OrgID As String,
Upgrade As Boolean) As tagMultiuserStatus

#### **Parameters**

### **OrgID**

[in] organization (database) ID of the database whose lock is being upgraded or downgraded

## Upgrade

[in] flag indicating whether the program's lock on the organization's database should be upgraded (TRUE) or downgraded (FALSE)

#### **Return Value**

Returns the status of the call.

## Remarks

If possible, avoid using this method. Lock manipulation should be done by views rather than by UIs. Lock manipulation (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet.

Before calling RegradeOrg, the program must have already locked the organization's database by calling LockOrg. Calls to RegradeOrg may be nested. If a program initially has shared access to the organization's database, nested calls to RegradeOrg to upgrade and then restore the program's lock on the organization's database will result in the database

being locked exclusively until the outermost level of nested calls restores the lock, returning it to shared access.

# AccpacMultiuser.Test

Tests if the specified resource is locked, and if it's locked, indicates whether the lock is exclusive or shared.

Function Test(
Resource As String,
Exclusive As Boolean) As Boolean

### **Parameters**

Resource

[in] the name of the resource

**Exclusive** 

[out] flag indicating whether the existing lock is exclusive (TRUE) or shared (FALSE), if the resource is locked

## **Return Value**

Returns whether the specified resource is locked.

## **Remarks**

If possible, avoid using this method. Locking should be done by views rather than by UIs. Locking (by the UI or macro) is especially dangerous when ACCPAC is running remotely over the Internet.

# AccpacMultiuser.UnlockApp

Unlocks an application's data that was locked by a previous call to LockApp.

Function UnlockApp(
OrgID As String,
AppID As String) As <u>tagMultiuserStatus</u>

### **Parameters**

OrgID

[in] the organization (database) ID of the company

**AppID** 

[in] two-letter application ID of the application whose data is to be unlocked

## **Return Value**

Returns the status of the call.

## **Remarks**

If possible, avoid using this method. However, if the application's data is currently locked by the UI (through an earlier <u>LockApp</u> call), use this method to unlock the data.

# AccpacMultiuser.UnlockOrg

Unlocks an organization's database locked by a previous call to LockOrg.

Function UnlockOrg(OrgID As String) As tagMultiuserStatus

### **Parameters**

OrgID

[in] the organization (database) ID of the database to be unlocked

### **Return Value**

Returns the status of the call.

### Remarks

If possible, avoid using this method. However, if the organization's database is currently locked by the UI (through an earlier <u>LockOrg</u> call), use this method to unlock the database.

# AccpacMultiuser.UnlockRsc

Unlocks a resource that was locked by a previous call to LockRsc.

Function UnlockRsc(Resource As String) As tagMultiuserStatus

### **Parameters**

Resource

[in] the name of the resource to unlock

### **Return Value**

Returns the status of the call.

## Remarks

If possible, avoid using this method. However, if the resource is currently locked by the UI (through an earlier <u>LockRsc</u> call), use this method to unlock the resource.

# **AccpacMultiuser Object**

The AccpacMultiuser object provides methods to control multi-user access to ACCPAC. Multi-user access is controlled by performing locking on predefined resources.

#### Remarks

Avoid using the AccpacMultiuser object if possible. Locking should be done by views rather than by UIs or macros. Locking is especially dangerous when ACCPAC is running remotely over the Internet.

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetMultiuser</u> property.

Locking can be done on the following resources:

- a specific resource identified by a resource string
- an application's data (identified by the organization ID of the database, together with the 2-letter application ID)
- an organization's data (identified by the organization ID of the database)

# AccpacOrganization.GetOrgInfo

Retrieves all the properties of an AccpacOrganization object in a single method call.

Sub GetOrgInfo(

DatabaseID As String,

Name As String,

SystemDatabaseID As String,

Type As tagOrganizationTypeEnum,

SecEnabled As Boolean)

### **Parameters**

#### DatabaseID

[out] returns the database (organization) ID, as defined in ACCPAC Database Setup

#### Name

[out] returns the database name, as defined in ACCPAC Database Setup

## SystemDatabaseID

[out] returns the database ID of the associated system database; returns "" if the current organization is a system database

## Туре

[out] returns the database type

#### SecEnabled

[out] returns whether security is enabled for the database

# AccpacOrganization.DatabaseID

Returns the organization (database) ID of the current organization.

Property DatabaseID As String (read-only)

# AccpacOrganization.Name

Returns the descriptive name of the database (organization).

Property Name As String (read-only)

# AccpacOrganization.SystemDatabaseID

Returns the organization (database) ID of the system database that the current organization belongs to, if it is a company database. If the organization itself is a system database, this property returns "".

Property SystemDatabaseID As String (read-only)

# **AccpacOrganization.Type**

Returns the database type.

Property Type As <u>tagOrganizationTypeEnum</u> (read-only)

# **Remarks**

Returns whether the organization is a company or system database.

# **AccpacOrganization Object**

The AccpacOrganization object provides details of an ACCPAC database set up in Database Setup.

### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacOrganizations</u> collection.

There are two kinds of databases that can be defined in ACCPAC:

- Company: A company database stores application data (covers most Common Services functions and all other applications). Each company database must be linked to a system database.
- System: A system database stores the data common to multiple company databases, such as currency information, user accounts, and authorization information. A system database can be associated with multiple company databases.

# AccpacOrganizations.GetOrgsInfo

Gets database (organization) information for all companies in one call.

Sub GetOrgsInfo(
Count As Long,
DatabaseIDs() As String,
Names() As String,

SystemDatabaseIDs() As String,

Types() As Integer,

SecEnabled() As Boolean)

### **Parameters**

#### Count

[out] returns the total number of company and system databases (count of items in the returned array, DatabaseIDs)

#### **DatabaseIDs**

[out] returns an array of company and system database IDs

#### Names

[out] returns an array where each item is the name of the organization associated with the corresponding item in *DatabaseIDs* 

## SystemDatabaseIDs 5 4 1

[out] returns an array where each item is either the associated system database ID, if the corresponding item in DatabaseIDs is a company database, or "", if the corresponding item in DatabaseIDs is a system database

## Types

[out] an array where each item is the database type that's associated with the corresponding item in *DatabaseIDs*. Each item in this array is actually a constant in <a href="mailto:tagDBLinkTypeEnum">tagDBLinkTypeEnum</a>.

# SecEnabled

[out] returns an array where each item indicates whether security is enabled for the corresponding item in *DatabaseIDs* 

# **AccpacOrganizations.Item**

Returns an <u>AccpacOrganization</u> object from the collection based on its organization ID.

Function Item(OrgID As String) As <u>AccpacOrganization</u> (*default*)

### **Parameters**

**OrgID** 

[in] the organization ID of the <u>AccpacOrganization</u> object in the collection

### **Return Value**

Returns the <u>AccpacOrganization</u> object representing the company specified by the organization ID.

### Remarks

Since the Item method is the default method of the collection, you do not have to specify "Item". For example, the following syntax forms are interchangeable:

Organizations.Item(*OrgID*)
Organizations(*OrgID*)

# AccpacOrganizations.ItemByIndex

Return the <u>AccpacOrganization</u> object of the collection based on its 0-based index in the collection.

Function ItemByIndex(Index As Long) As AccpacOrganization

### **Parameters**

Index

[in] the index of the <u>AccpacOrganization</u> object in the 0-based collection

### **Return Value**

An AccpacOrganization object.

# **AccpacOrganizations.Next**

Retrieves an object that represents a company set up in the system.

Function Next(pVal As <u>AccpacOrganization</u>) As Boolean

## **Parameters**

pVal

[out] an AccpacOrganization object, which represents a company set up in the system

#### **Return Value**

Returns whether information on the next company is available.

### **Remarks**

If Next is called after calling <u>Reset</u>, the first <u>AccpacOrganization</u> object in the collection is retrieved. Subsequent calls to Next would navigate through the collection and return the next <u>AccpacOrganization</u> object.

# **AccpacOrganizations.Reset**

Re-reads database setup information from the system, and resets the current position to the beginning of the collection. A subsequent call to <a href="Next">Next</a> would retrieve the first <a href="AccpacOrganization">AccpacOrganization</a> object (the first company) in the collection.

Sub Reset()

# **AccpacOrganizations.Count**

Returns the number of <u>AccpacOrganization</u> objects in the collection.

Property Count As Long (read-only)

# **Remarks**

Returns a count of all company and system databases defined in ACCPAC Database Setup.

# **AccpacOrganizations Object**

The AccpacOrganizations object represents a collection of all the organizations set up in ACCPAC. It exposes a collection of AccpacOrganization objects.

### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from The <u>AccpacSession</u> object's <u>Organizations</u> property.

# AccpacPrintSetup.Query

Displays the local Windows printer setup dialog that allows the user to select printer settings.

Function Query(hWnd As Long) As Boolean

### **Parameters**

hWnd

[in] window handle of the caller

#### **Return Value**

Returns whether printer settings have been changed by the user.

### **Remarks**

If Query returns TRUE, the <u>Save</u> method should be called to save the changes for the current user.

The AccpacPrintSetup object should only be accessed when ACCPAC is running locally (when the associated <u>AccpacSession</u> object's <u>IsRemote</u> property is FALSE).

# AccpacPrintSetup.Save

Saves the local printer setup information for the current user.

Sub Save()

### **Remarks**

The Save method is usually called after <u>Query</u> (which shows the printer setup dialog) returns TRUE, indicating that printer setup information has changed.

The AccpacPrintSetup object should only be accessed when ACCPAC is running locally (when the associated <u>AccpacSession</u> object's <u>IsRemote</u> property is FALSE).

# **AccpacPrintSetup.Destination**

Returns/sets the default print destination for ACCPAC reports.

Property Destination As <a href="mailto:tagPrintDestinationEnum">tagPrintDestinationEnum</a>

### **Remarks**

Use the Destination property of the <u>AccpacReport</u> object if you want a particular report to override the user's default print settings.

# AccpacPrintSetup.DeviceName

Returns/sets the device name of the printer.

Property DeviceName As String

# AccpacPrintSetup.DriverName

Returns/sets the filename of the printer driver.

Property DriverName As String

# **AccpacPrintSetup.Duplex**

Returns/sets whether to use duplex printing.

Property Duplex As Boolean

# **AccpacPrintSetup.Orientation**

Returns/sets portrait or landscape print orientation.

**Property Orientation As Long** 

# AccpacPrintSetup.OutputName

Returns/sets the device name for the physical output medium. This is usually the name of the printer port, such as LPT1.

Property OutputName As String

# **AccpacPrintSetup.PaperSize**

Returns/sets the paper size.

Property PaperSize As Long

# AccpacPrintSetup.PaperSource

Returns/sets the paper source.

Property PaperSource As Long

# AccpacPrintSetup.PrintDir

Returns/sets the output directory, if the print destination is a file.

Property PrintDir As String

### **Remarks**

Use the PrintDir property of the <u>AccpacReport</u> object if you want a particular report to override the user's default print settings.

# **AccpacPrintSetup Object**

The AccpacPrintSetup object stores the printer settings used for reporting, and it provides facilities to query the user and save default print settings when ACCPAC is run from the local machine.

### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetPrintSetup</u> method.

Many of the AccpacPrintSetup object's (print setting) properties have corresponding properties in the <u>AccpacReport</u> object. Set the print-related properties directly in the AccpacReport object (rather than in the AccpacPrintSetup object) if you want a particular report's settings to override the user's default print settings.

# AccpacProperty.PropClear

Clears the property from the user's property file.

Function PropClear() As Long

# **Return Value**

Returns 0 upon success and a non-zero error code upon failure.

### Remarks

The property whose value is being cleared is specified by this AccpacProperty object's <u>Keyword</u>, <u>MenuID</u>, and <u>objectID</u>.

# AccpacProperty.PropGet

Retrieves the ACCPAC property value based on the values set in the AccpacProperty object's <u>Keyword</u>, <u>MenuID</u> and <u>objectID</u>. The property value can be retrieved as a String or as an array of Bytes.

```
Function PropGet(
varBuf as Variant,
[PropType As tagPropertyType]) As Long
```

#### **Parameters**

```
varBuf
  [out] returns the property value

PropType
  [in, optional] the property value's type
```

### **Return Value**

Returns 0 upon success and a non-zero error code upon failure.

### **Remarks**

If PropType is set as PROP\_AS\_STRING, the property is returned as a String. If PROP\_AS\_ARRAY is specified, the property is returned as an array of Bytes. If PropType isn't specified, the property is returned as a String.

# AccpacProperty.PropGetEx

Retrieves the ACCPAC property corresponding to the specified ObjectID, MenuID and Keyword (as opposed to the AccpacProperty object's own ObjectID, MenuID, and Keyword). The property value can be retrieved as a String or as an array of Bytes.

```
Function PropGetEx(
objectID As String,
menuID As String,
keyword As String,
Buffer As Variant,
PropType As tagPropertyType) As Long
```

#### **Parameters**

```
objectID
  [in] the object (program) ID of the property

menuID
  [in] the menu ID of the property

keyword
  [in] the keyword of the property

Buffer
  [out] returns the property value

PropType
  [in] the property value's type
```

### **Return Value**

Returns 0 upon success and a non-zero error code upon failure.

If PropType is set as PROP\_AS\_STRING, the property is returned as a String. If PROP\_AS\_ARRAY is specified, the property is returned as an array of Bytes. Unlike the <a href="PropGet">PropGet</a> method, you must specify PropType for PropGetEx.

# AccpacProperty.PropPut

Saves the property value to the user's property file.

```
Function PropPut(
varBuf As Variant,
lSize As Long) As Long
```

### **Parameters**

```
varBuf
[in] the property valueISize
[in] the size of the property value passed in through varBuf
```

### **Return Value**

Returns 0 upon success and a non-zero error code upon failure.

## **Remarks**

The property value passed into the function must be a String or an array of Bytes. The size of the property value is the length of the string or the number of elements in the array of Bytes.

The property whose value is being saved is specified by this AccpacProperty object's <u>Keyword</u>, <u>MenulD</u>, and <u>objectID</u>.

# AccpacProperty.AppID

Returns/sets the application ID of the property.

Property AppID As String

# **AccpacProperty.AppVersion**

Returns/sets the application version of the property.

Property AppVersion As String

# AccpacProperty.Keyword

Returns/sets the keyword of the property.

Property Keyword As String

# AccpacProperty.MenuID

Returns/sets the menu ID of the property.

Property MenuID As String

# **AccpacProperty.objectID**

Returns/sets the object (program) ID of the property.

Property objectID As String

## **Remarks**

For a UI, the object ID is the UI's Roto ID.

## **AccpacProperty Object**

The AccpacProperty object provides access to application properties stored in System Manager. With an AccpacProperty object, UIs, macros, and other components can store run time properties in System Manager. These properties are available in later sessions and are machine-independent.

ACCPAC properties are stored on a per-user basis and each property is identified by the combination of:

- Program (Object) ID the Roto ID of the UI
- Menu ID the menu ID
- Keyword application-defined keyword of the property

Property values can be stored as strings or as arrays of binary values.

### Remarks

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>GetAccpacProperty</u> method.

## AccpacReport.CompleteProcessServerSe

Takes the AccpacProcessServerSetup object for the report and uses it to complete the report configuration. Call this after the AccpacProcessServerSetup object has been configured and before printing.

Function CompleteProcessServerSettings() As Long

#### **Return Value**

Returns 0 upon success and a non-zero error code upon failure.

### Remarks

If a report is configured to be run through the Process Server, the following must be done:

- 1. Create the report object by calling ReportSelect of an AccpacSession object.
- 2. Check the RequiresProcessServerSettings property.
- 3. If TRUE, call <u>GetProcessServerSetup</u> to get an AccpacProcessServerSetup object.
- 4. Configure the AccpacProcessServerSetup object.
- 5. Call CompleteProcessServerSettings to save the AccpacProcessServerSetup settings to the report object.
- 6. Call **Confirm** as usual.

The AccpacProcessServerSetup object is a hidden member of the ACCPAC COM API library.

# **AccpacReport.Confirm**

Sets up the print settings for the report and loads the saved print settings, if any, for the current report. Optionally, based on the *ShowDialog* parameter, a print settings/confirmation dialog is displayed. This dialog allows the user to change print settings.

Function Confirm(
ShowDialog As Boolean,
hWnd As Long) As Boolean

#### **Parameters**

ShowDialog
[in] whether to show a print setup/confirmation dialog

hWnd
[in] window handle of the caller

### **Return Value**

Returns whether or not the user chose to proceed with printing the report.

### Remarks

If the current session is a remote connection to an ACCPAC server, the <u>print destination</u> is always set to <u>PD\_PREVIEW</u> and no print settings/confirmation dialog is displayed as the ShowDialog parameter is ignored.

# AccpacReport.GetProcessServerSetup

Returns an AccpacProcessServerSetup object for the report.

Function GetProcessServerSetup() As AccpacProcessServerSetup

#### **Return Value**

Returns the AccpacProcessServerSetup object for the report.

#### Remarks

If a report is configured to be run through the Process Server, the following must be done:

- 1. Create the report object by calling ReportSelect of an AccpacSession object.
- 2. Check the RequiresProcessServerSetup property.
- 3. If TRUE, call GetProcessServerSetup to get an AccpacProcessServerSetup object.
- 4. Configure the AccpacProcessServerSetup object.
- 5. Call <u>CompleteProcessServerSettings</u> to save the AccpacProcessServerSetup settings to the report object.
- 6. Call Confirm as usual.

The AccpacProcessServerSetup object is a hidden member of the <u>ACCPAC COM API</u> library.

## **AccpacReport.PrinterSetup**

Configures the print settings for the report according to the settings in the supplied <a href="AccpacPrintSetup">AccpacPrintSetup</a> object.

Sub PrinterSetup(pSetup As <a href="AccpacPrintSetup">AccpacPrintSetup</a>)

### **Parameters**

pSetup
[in] an AccpacPrintSetup object that contains the desired print
settings

## **Remarks**

In most cases, a report can make use of the current user's default print settings, or changes can be made to some common print settings by setting report properties, such as <u>Destination</u>, and <u>NumOfCopies</u>.

PrinterSetup is only called if the report needs some specific print settings to be set, such as the printer's device name or the report's orientation.

## AccpacReport.PrintReport

Prints a report to the printer, to a file, to email, or to the preview screen, depending on the value of the Destination property.

Function PrintReport() As Boolean

#### **Return Value**

Returns whether a web-based report was generated.

### Remarks

If the return value is TRUE, a web-based report was generated. This type of report is generated when the current session is accessing a remote ACCPAC application server. In this case, applications should access the WebReportURL property (which stores the URL where the report can be retrieved) and display the report accordingly.

If the return value is FALSE, the session is in-process to the caller, and the report has been printed to the desired destination.

### **Remarks**

For reports that include parameters, make sure that <u>SetParam</u> calls have been made before calling PrintReport.

# AccpacReport.ReInit

Re-initializes the <u>AccpacReport</u> object so that it may be used to print a new report.

Sub ReInit()

## Remarks

Use this method to use the same report object to print multiple instances of a report, each with different report parameters.

# AccpacReport.SaveWebReportToLocal

Downloads a web-based report from the remote ACCPAC server to a local file. The report is generated as the specified report output type.

Function SaveWebReportToLocal(
ReportType As <a href="mailto:tagReportOutputType">tagReportOutputType</a>) As String

### **Parameters**

ReportType
[in] the report output type to generate

### **Return Value**

Returns the path to the local report file (including the filename).

# AccpacReport.SetParam

Sets a value to the specified report parameter name.

Function SetParam(
ParamName As String,
ParamValue As String) As Boolean

### **Parameters**

#### **ParamName**

[in] the report parameter name (must be valid parameter defined for the current report)

#### **ParamValue**

[in] the report parameter's value

### **Return Value**

Returns whether the parameter was successfully set. The function returns FALSE if the specified parameter name is not recognized as a valid parameter defined for the report.

### Remarks

SetParam calls should be made before PrintReport is called.

# **AccpacReport.Collate**

Returns/sets whether the print output will be collated, if the number of copies is set to more than one.

Property Collate As Boolean

## **Remarks**

This property is ignored if the print destination is not Printer (PD\_PRINTER).

# **AccpacReport.Destination**

Returns/sets the print destination of the report.

Property Destination As <a href="mailto:tagPrintDestinationEnum">tagPrintDestinationEnum</a>

# **AccpacReport.Format**

Returns/sets the output format of the report.

Property Format As <a href="mailto:tagPrintFormatEnum">tagPrintFormatEnum</a>

## **Remarks**

Currently, this property is only valid for an email output destination. It is ignored for all other print destinations.

# AccpacReport.Name

Returns the name of the report.

Property Name As String (read-only)

## **Remarks**

The report's name is set when the report is created by calling the <u>AccpacSession</u> object's <u>ReportSelect</u> method.

# AccpacReport.NumOfCopies

Returns/sets the number of copies to print.

Property NumOfCopies As Long

## **Remarks**

This property is only valid if the <u>print destination</u> is Printer. It is ignored for all other print destinations.

# **AccpacReport.OutputDestination**

Returns/sets whether the report will be generated on the remote ACCPAC server or on the local client.

Property OutputDestination As <a href="mailto:tagOutputDestinationEnum">tagOutputDestinationEnum</a>

### **Remarks**

This property only applies when the client is connecting to a remote server and is ignored otherwise.

# AccpacReport.PrintDir

Returns/sets the output directory, if the <u>print destination</u> is set to File. It is ignored for all other print destinations.

Property PrintDir As String

### Remarks

This property should be set as the full path to the output directory.

# RequiresProcessServerSettings

Returns whether the report must be configured for the Process Server before printing.

Property RequiresProcessServerSettings As Boolean (*read-only*)

### **Remarks**

If this property is TRUE, then the following must be done before calling Confirm:

- 1. Call <u>GetProcessServerSetup</u> to get an AccpacProcessServerSetup object.
- 2. Configure the AccpacProcessServerSetup object.
- 3. Call <u>CompleteProcessServerSettings</u> to save the AccpacProcessServerSetup settings to the report object.

The AccpacProcessServerSetup object is a hidden member of the <u>ACCPAC COM API</u> library.

# AccpacReport.UseProcessServer

Returns whether the Process Server is to be used to print the report.

Property UseProcessServer As Boolean (read-only)

## AccpacReport.WebReportURL

Returns the URL of the generated web-based report (<u>PrintReport</u> will return TRUE if a web report has been generated).

Property WebReportURL As String (read-only)

### **Remarks**

This property always returns an empty string ("") before <u>PrintReport</u> is called. If PrintReport is TRUE, a web report has been generated, and the generated report should be shown by calling an AccpacPrintReport object's ReportViewer method.

The web report URL is only valid during the lifetime of the current Report object. The report is removed from the web server when the object is destroyed, or when Relnit is called.

# **AccpacReport Object**

The AccpacReport object provides methods to generate an ACCPAC report.

### Remarks

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacSession</u> object's <u>ReportSelect</u> method.

## **Example**

The following Visual Basic code illustrates how to use the methods and properties of the AccpacReport object.

```
Dim Session As AccpacCOMAPI.AccpacSessionSet Session =
CreateObject("ACCPAC.AccpacSession")Session.Open "ADMIN", "ADMIN",
"SAMLTD", Date, 0, ""
Dim rpt As AccpacCOMAPI.AccpacReport
Set rpt = Session.ReportSelect("CSCURRATES", " ", " ")
rpt.SetParam "FROMCUR", " "
rpt.SetParam "TOCUR", "ZZZ"
rpt.SetParam "FROMTYPE", " "
rpt.SetParam "TOTYPE", "ZZ"
rpt.SetParam "FROMDATE", Format(Date, "yyyymmdd")
rpt.SetParam "TODATE", Format(Date, "yyyymmdd")
rpt.PrintDestination = PD_PREVIEW
rpt.PrintReport
```

# AccpacSession.CheckHomeCurrency

Returns whether the home currency defined in the Company Profile UI of ACCPAC Common Services for the logged in company exists in the company's currency codes table.

Function CheckHomeCurrency() As Boolean

### **Return Value**

Returns TRUE if the company's home currency exists in the currency codes table; otherwise, returns FALSE.

The home currency is also referred to as the functional currency.

# **AccpacSession.CheckReminders**

Returns whether there are active reminders for the currently logged in user.

Function CheckReminders() As Boolean

### **Return Value**

Returns TRUE if the logged in user has active reminders; otherwise returns FALSE.

# AccpacSession.CheckRestartRecs

Returns whether restart records exist for the logged in company in the current session.

Sub CheckRestartRecs(
RestartRecsExist As Boolean,
IsAdmin As Boolean)

### **Parameters**

RestartRecsExist [out] returns whether restart records exist

**IsAdmin** 

[out] if RestartRecsExist is TRUE, returns whether or not the current user is ADMIN.

# AccpacSession.CheckSessionDate

Checks if the session date is a valid date defined in the company's fiscal calendar, and whether the fiscal period in which the session date belongs is active.

Sub CheckSessionDate(
DateInFiscal As Boolean,
PeriodActive As Boolean)

#### **Parameters**

#### DateInFiscal

[out] returns whether the session date is a date defined in the fiscal calendar

### PeriodActive

[out] if *DateInFiscal* is TRUE, returns whether the fiscal period, to which the session date belongs, is active

# AccpacSession.CheckSessionDate2

Checks if the session date is a valid date defined in the company's fiscal calendar, whether the fiscal period in which the session date belongs is active and open (not locked), and whether a warning should be displayed if the period is locked.

Sub CheckSessionDate2(
DateInFiscal As Boolean,
PeriodActive As Boolean,
PeriodOpen As Boolean,
WarnPeriodLocked As Boolean)

### **Parameters**

#### DateInFiscal

[out] returns whether the session date is a date defined in the fiscal calendar

#### PeriodActive

[out] if *DateInFiscal* is TRUE, returns whether the fiscal period to which the session date belongs is active

### PeriodOpen

[out] if *DateInFiscal* is TRUE, returns whether the fiscal period to which the session date belongs is open (not locked)

#### WarnPeriodLocked

[out] returns whether a warning should be displayed if the fiscal period to which the session date belongs is locked (if *PeriodOpen* is FALSE)

# **AccpacSession.Clone**

Clones the current <u>AccpacSession</u> object. The new object has the same authentication information as the current session object, but is attached to a different company database ID.

Function Clone(
CompanyID As String) As AccepacSession

### **Parameters**

CompanyID

[in] the database ID of the company to which the new session should be attached

### **Return Value**

Returns the cloned **AccpacSession** object.

# AccpacSession.Close

Closes the session and releases all resources allocated to the session.

Sub Close()

### **Remarks**

When a session is closed, the user is signed off from the company, and all connections to the company database are closed. All objects that are created from this session object become invalid and should not be accessed again.

The session closes automatically when the <u>AccpacSession</u> object is destroyed.

## AccpacSession.CreateObjectHandle

Creates a new object handle to be used to launch another application. The method is also used to obtain the CLSID and codebase of the application to be launched. When an application is launched using the newly created object handle, it inherits the session information from the current session (includes authentication information, and optionally, application data).

Sub CreateObjectHandle(
ObjectID As String,
ObjectKey As String,
ObjectHandle As String,
CLSID As String,
Codebase As String)

### **Parameters**

## ObjectID

[in] the Roto ID of the application object to be launched

## **ObjectKey**

[in] a key to be passed to the launched object The key is a String of any format as long as it is understood by both the current application and the object to which the key is passed.

## ObjectHandle

[out] returns the newly created object handle

#### CLSID

[out] returns the COM class ID (CLSID) of the application object

#### Codebase

[out] returns the codebase of the object

If the session is local, the codebase is the file path to the

- object's locally installed COM object.
- If the session is accessing a remote ACCPAC server, the codebase is the URL of the application object's CAB file

### **Remarks**

This method creates an object handle, but does not launch the application identified by the Roto ID. The caller is responsible for launching the desired object in an appropriate way, passing along the object handle which can be used by that object to initialize and open its own session.

The application to be launched is specified by its Roto ID. The system looks up the CLSID and generates the codebase of the application object based on this Roto ID.

The *ObjectKey* is a string passed to the launched application. The most common use of this is to pass the key of the initial record to display on the launched application.

# AccpacSession.CreateObjectHandle2

Creates a new object handle to be used to launch another application.

Function CreateObjectHandle2( objectID As String, ObjectKey As String) As String

### **Parameters**

objectID

[in] the Roto ID of the application object to be launched

ObjectKey

[in] a key to be passed to the launched object The key is a String of any format as long as it is understood by both the current application and the object to which the key is passed.

## **Return Value**

Returns the new object handle.

## **Remarks**

Script languages, such as Javascript, which do not support "out" parameters should use this version of the <a href="CreateObjectHandle">CreateObjectHandle</a> method.

# AccpacSession.CreateProfile

Creates a new UI Customization Profile with the specified profile ID and description in ACCPAC's Administrative Services.

Sub CreateProfile(
ProfileID As String,
ProfileDesc As String)

## **Parameters**

ProfileID
[in] the profile ID

ProfileDesc
[in] the profile description

### Remarks

An administrator (ADMIN) would normally create a new profile using the UI Profile Maintenance UI in ACCPAC's Administrative Services. This UI can either be accessed directly or through the Customize menu option.

## AccpacSession.GetAccpacMeter

Creates and returns an AccpacMeter object which provides information about the progress of any lengthy superview calls in progress for the session and program, such as the percent completed.

The AccpacMeter object is a hidden member of the ACCPAC COM API library. It is provided mainly for ACCPAC controls to display progress information. Application programs do not normally need to access this object directly.

Sub GetAccpacMeter(pVal As AccpacMeter)

### **Parameters**

pVal [out] returns an AccpacMeter object

## **Return Value**

Returns an AccpacMeter object attached to the current session.

# AccpacSession.GetAccpacProperty

Creates and returns an <u>AccpacProperty</u> object. The AccpacProperty object provides access to ACCPAC run time properties that are stored for the current user.

Function GetAccpacProperty() As <u>AccpacProperty</u>

### **Return Value**

Returns an AccpacProperty object that stores properties for the current user.

# AccpacSession.GetAppDependenciesFor

Returns a list of dependent classes based on the specified application ID, program name, version, and language code.

Sub GetAppDependenciesForLanguage(
AppID As String,
PgmName As String,
AppVersion As String,
Language As String,
CodebaseType As tagDistFileType,
CLSIDs() As String,
Codebases() As String)

#### **Parameters**

```
AppID
[in] the two-letter application ID

PgmName
[in] the Roto ID of the application

AppVersion
[in] the application version

Language
[in] the 3-letter language code

CodebaseType
[in] the format of the codebases to retrieve

CLSIDs
[out] returns an array of COM class IDs for the dependent classes

Codebases
[out] returns an array of codebases for the dependent classes
```

### **Remarks**

The format of the returned items in the Codebases array depends on the value of CodebaseType:

- DF\_DYNAMIC Depends on the client's connection. If the client is remote, the codebases are the URLs to the dependent classes' CAB files. If the client is in-process, the codebases are the local paths (including filenames) to the classes' (OCX, DLL or EXE) files.
- DF\_REMOTE Codebases are always returned as the URLs to the dependent classes' CAB files.
- DF\_LOCAL Codebases are always returned as the local paths (including filenames) to the dependent classes' (OCX, DLL or EXE) files.

To get the dependent classes based on the current session's ProgramName and the current user's language, use <u>GetDependencies</u> instead.

# AccpacSession.GetCurRateTypeDescripti

Retrieves the description of the specified currency rate type code.

Function GetCurRateTypeDescription(
rateTypeCode As String,
rateTypeDescription As String) As Boolean

## **Parameters**

rateTypeCode
 [in] the rate type code

rateTypeDescription
 [out] returns the description of the supplied rate type code

### **Return Value**

Returns whether the specified currency rate type code exists in the Currency Rate Types table.

# AccpacSession.GetCurRateTypeDescripti

Retrieves the description of the specified currency rate type code.

Function GetCurRateTypeDescription2( rateTypeCode As String, rateTypeDescription) As Boolean

### **Parameters**

rateTypeCode
 [in] the rate type code

rateTypeDescription
 [out] returns the description of the supplied rate type code

### **Return Value**

Returns whether the specified currency rate type code exists in the Currency Rate Types table.

## Remarks

Script languages, such as Javascript, which do not support "out" parameters, should use this variation of the <u>GetCurRateTypeDescription</u> function.

# **AccpacSession.GetCurrency**

Creates and returns an <u>AccpacCurrency</u> object that contains information about the specified currency code.

Function GetCurrency(CurrencyCode As String) As AccpacCurrency

### **Parameters**

CurrencyCode [in] the currency code to retrieve

### **Return Value**

Returns an <u>AccpacCurrency</u> object that represents the specified currency code.

## AccpacSession.GetCurrencyRate

Creates and returns an <u>AccpacCurrrencyRate</u> object that represents the exchange rate between the specified home and source currency codes of the specified rate type on a certain date.

Function GetCurrencyRate(
HomeCurrencyCode As String,
RateType As String,
SourceCurrencyCode As String,
Date As Date) As AccpacCurrencyRate

## **Parameters**

HomeCurrencyCode
[in] the home currency code

RateType
[in] the currency rate type code

SourceCurrencyCode
[in] the source currency code

## Return Value

Date

Returns an <u>AccpacCurrrencyRate</u> object that contains details of the exchange rate.

[in] the date for which the exchange rate is retrieved

## Remarks

The home currency is also referred to as the functional currency.

# AccpacSession.GetCurrencyRateCompos

Creates and returns an <u>AccpacCurrencyRate</u> object that represents the composite rate between the specified home and source currency codes of the specified rate type on a certain date.

Function GetCurrencyRateComposite(
HomeCurrencyCode As String,
RateType As String,
SourceCurrencyCode As String,
Date As Date) As AccpacCurrencyRate

### **Parameters**

HomeCurrencyCode [in] the home currency code

RateType

[in] the currency rate type code

SourceCurrencyCode

[in] the source currency code

Date

[in] the date for which the exchange rate is retrieved

### **Return Value**

Returns an <u>AccpacCurrencyRate</u> object that contains details of the composite exchange rate.

## Remarks

If both currencies are non-block currencies, this method returns the same rate as <u>GetCurrencyRate</u>.

The home currency is also referred to as the functional currency.

# AccpacSession.GetCurrencyRateFloating

Creates and returns an <u>AccpacCurrencyRate</u> object that represents the floating exchange rate between the specified home and source currency codes of the specified rate type on a certain date.

Function GetCurrencyRateFloating(
HomeCurrencyCode As String,
RateType As String,
SourceCurrencyCode As String,
Date As Date) As AccpacCurrencyRate

### **Parameters**

HomeCurrencyCode [in] the home currency code

RateType

[in] the currency rate type code

SourceCurrencyCode

[in] the source currency code

Date

[in] the date for which the exchange rate is retrieved

### **Return Value**

Returns an <u>AccpacCurrencyRate</u> object that contains details of the floating exchange rate.

## Remarks

If both currencies are non-block currencies, this method returns the same rate as <u>GetCurrencyRate</u>.

The home currency is also referred to as the functional currency.

# AccpacSession.GetCurrencyTable

Creates and returns an <u>AccpacCurrencyTable</u> object that represents the currency table set up in ACCPAC Common Services with the specified code and rate type.

Function GetCurrencyTable(
CurCode As String,
RateType As String) As <u>AccpacCurrencyTable</u>

### **Parameters**

CurCode
[in] the currency code

RateType
[in] the currency rate type code

## **Return Value**

Returns an <u>AccpacCurrencyTable</u> object that contains details of the currency table with the specified code and rate type.

# AccpacSession.GetDBLink

A private method.

Function GetDBLink(
LinkType As tagDBLinkTypeEnum,
Flags As tagDBLinkFlagsEnum) As AccpacDBLink

### **Parameters**

```
LinkType
[in] the type of database link to retrieve

Flags
[in] the type of access mode to place on the database link
```

### **Return Value**

Returns an AccpacDBLink object that represents the database link.

## Remarks

## **AccpacSession.GetDependencies**

Returns a list of dependent classes based on the current session's <a href="Program Name">Program Name</a> and the user's language setting (the current session's <a href="UserLanguage">UserLanguage</a>).

```
Sub GetDependencies(
CodebaseType As tagDistFileType,
CLSIDs() As String,
Codebases() As String)
```

#### **Parameters**

CodebaseType
[in] format of the codebases to retrieve

**CLSIDs** 

[out] returns an array of COM class IDs for the dependent classes

Codebases

[out] returns an array of codebases for the dependent classes

### **Remarks**

The format of the returned items in the Codebases array depends on the value of CodebaseType passed in:

- DF\_DYNAMIC Depends on the client's connection. If the client is remote, the codebases are the URLs to the dependent classes' CAB files. If the client is in-process, the codebases are the local paths (including filenames) to the classes' (OCX, DLL or EXE) files.
- DF\_REMOTE Codebases are always returned as the URLs to the dependent classes' CAB files.
- DF\_LOCAL Codebases are always returned as the local paths

(including filenames) to the dependent classes' (OCX, DLL or EXE) files.

To get the dependent classes for a language other than the user's language, use <u>GetDependenciesForLanguage</u> instead.

## AccpacSession.GetDependenciesForLang

Returns a list of dependent classes based on the current session's <a href="Program Name">Program Name</a> and the specified language code.

Sub GetDependenciesForLanguage(
Language As String,
CodebaseType As tagDistFileType,
CLSIDs() As String,
Codebases() As String)

### **Parameters**

Language

[in] the three-letter language code

CodebaseType

[in] the format of the codebases to retrieve

**CLSIDs** 

[out] returns an array of COM class IDs for the dependent classes

Codebases

[out] returns an array of codebases for the dependent classes

## Remarks

The format of the returned items in the Codebases array depends on the value of CodebaseType passed in:

- DF\_DYNAMIC Depends on the client's connection. If the client is remote, the codebases are the URLs to the dependent classes' CAB files. If the client is in-process, the codebases are the local paths (including filenames) to the classes' (OCX, DLL or EXE) files.
- DF\_REMOTE Codebases are always returned as the URLs to the

dependent classes' CAB files.

• DF\_LOCAL - Codebases are always returned as the local paths (including filenames) to the dependent classes' (OCX, DLL or EXE) files.

To get the dependent classes for the current user's language, use <u>GetDependencies</u> instead.

# AccpacSession.GetIniFileKey

Retrieves a value specified in the application's INI file.

Function GetIniFileKey(
AppID As String,
PrimaryKey As String,
SecondaryKey As String,
KeyData As String) As Boolean

### **Parameters**

### **AppID**

[in] the two-letter application ID

## PrimaryKey

[in] the primary key, which corresponds to the name of the section where the value is defined

## SecondaryKey

[in] the key name of the value, which corresponds to the name of the key where the value is defined

## KeyData

[out] returns the key value from the INI file, or the section name if an empty string ("") is passed in as the SecondaryKey

## **Return Value**

Returns whether or not the specified keys are found in the application INI file. If *SecondaryKey* is an empty string, returns whether or not the section is found.

## AccpacSession.GetIniFileKey2

Retrieves a value specified in the application's INI file.

Function GetIniFileKey2(
AppID As String,
PrimaryKey As String,
SecondaryKey As String,
KeyData) As Boolean

### **Parameters**

### **AppID**

[in] the two-letter application ID

## PrimaryKey

[in] the primary key, which corresponds to the name of the section where the value is defined

## SecondaryKey

[in] the key name of the value, which corresponds to the name of the key where the value is defined

## KeyData

[out] returns the key value from the INI file, or the section name if an empty string ("") is passed in as the *SecondaryKey* 

## **Return Value**

Returns whether or not the specified keys are found in the application INI file. If *SecondaryKey* is an empty string, returns whether or not the section is found.

## Remarks

Script languages, such as Javascript, which do not support "out" parameters should use this variation of the <u>GetIniFileKey</u> function.

## AccpacSession.GetInstalledReports

Retrieves a list of reports available from the specified application.

Function GetInstalledReports(AppID As String) As String()

### **Parameters**

AppID [in] the two-letter application ID

### **Return Value**

This method returns an array of Strings. This array contains a list of reports available from the application. Note that the report names do not include file paths.

### Remarks

This method accepts the application ID, but not the application version. The application version is automatically determined and is the version of the application activated for the current company.

The following locations are searched for report files:

- The application's language-specific directory, depending on the logged in user's language preference.
- Customization directory, if defined, for all users for the current logged in company.
- Customization directory, if defined, for all companies for the current logged in user.
- Customization directory, if defined, for all users for all companies.

The AccpacUI template provides a "Remote Report Selector" popup form

(RemoteRptSelector.frm) that can be added to VB UI projects. This popup form calls GetInstalledReports and displays the retrieved report list in a combo box. When the popup is closed, it returns the selected report name to the UI.

# AccpacSession.GetMultiuser

Creates and returns an <u>AccpacMultiuser</u> object that provides facilities to control multi-user access.

Function GetMultiuser() As <u>AccpacMultiuser</u>

### **Return Value**

Returns an <u>AccpacMultiuser</u> object that provides facilities to control multiuser access.

### Remarks

Avoid using the <u>AccpacMultiuser</u> object if possible. Locking should be done at the view level rather than at the UI level. Locking is especially dangerous when ACCPAC is running remotely over the Internet.

## AccpacSession.GetObjectCLSID

Retrieves the COM class ID (CLSID) and codebase of the specified application object.

Function GetObjectCLSID(
objectID As String,
DistFileType As tagDistFileType,
CLSID As String,
Codebase As String) As Boolean

### **Parameters**

```
objectID
  [in] the Roto ID of the application object

DistFileType
  [in] the format of the codebase to retrieve

CLSID
  [out] returns the COM class ID of the object

Codebase
  [out] returns the codebase of the object
```

## **Return Value**

Returns whether the specified object ID (Roto ID) is valid.

## Remarks

This method locates the CLSID of the specified object in the application's Roto file.

The format of the codebase returned depends on the value of

## DistFileType passed in:

- DF\_DYNAMIC Codebase depends on the client's connection. If the client is remote, the codebase is the URL of the object's CAB file. If the client is in-process, the codebase is the path to the OCX file (including filename).
- DF\_REMOTE Codebase is always returned as the URL to the object's CAB file.
- DF\_LOCAL Codebase is always returned as the local path to the application's OCX file (including filename).

# AccpacSession.GetObjectCLSID2

Retrieves the COM class ID (CLSID) and codebase of the specified application object.

Function GetObjectCLSID2(
objectID As String,
DistFileType As tagDistFileType) As String

### **Parameters**

objectID
 [in] the Roto ID of the application object
DistFileType
 [in] the format of the codebase to retrieve

### **Return Value**

Returns a string of the form <CLSID>@<Codebase>. Returns an empty string if objectID is not valid.

## **Remarks**

Script programs, such as Javascript, which do not support "out" parameters, should use this variation of the <u>GetObjectCLSID</u> function.

## AccpacSession.GetObjectKey

Retrieves the object key associated with the current session.

Function GetObjectKey() As String

### **Return Value**

Returns the object key associated with the object handle (set by the caller or parent application) that opened the current session.

Returns empty if the object key is not set.

### **Remarks**

An object key is associated with an object handle when the calling application calls <a href="CreateObjectHandle">CreateObjectHandle</a> with the key, and obtains an object handle. When the object handle is subsequently passed to another application and used to initialize a session, the key is tied to the newly opened session and can be obtained from GetObjectKey.

## AccpacSession.GetPrintSetup

Creates and returns an <u>AccpacPrintSetup</u> object that allows applications to control default printer settings.

If printer settings have been saved for the specified *menuID* and *ProgramID*, the returned object contains the saved settings. Otherwise, the object contains default print settings.

Function GetPrintSetup(
menuID As String,
ProgramID As String) As <u>AccpacPrintSetup</u>

### **Parameters**

#### menuID

[in] the menu ID used to identify the print property
Menu IDs are assigned by the system and applications do not have
control over the menu ID. If a specific menu ID is not known, an
application should pass in the object ID or an empty string (""). In
these cases, the AccpacSession object will use the system-assigned
menu ID.

## **ProgramID**

[in] the Roto ID of the application

The AccpacPrintSetup object uses the ProgramID and menu ID to locate default printer settings saved in the current user's properties file.

## **Return Value**

Returns the <u>AccpacPrintSetup</u> object.

## Remarks

This function is not available if the current session accesses a remote ACCPAC server. Therefore, only call this method if ACCPAC is running locally - when the session's <a href="Issaercota">Issaercota</a> property is FALSE.

After obtaining the AccpacPrintSetup object, you would call its <u>Query</u> method to show the printer setup dialog, then call its <u>Save</u> method (if any printer settings have changed).

# **AccpacSession.GetProfileCustomizations**

Retrieves UI customization settings (an array of hidden UI controls) for the specified *ProfileID* and screen identifier (*UIKey*).

Function GetProfileCustomizations(
ProfileID As String,
UIKey As String) As String()

### **Parameters**

ProfileID
[in] the profile ID

UIKey
[in] the unique key (identifier) for the UI
This UI must be part of the application whose ApplD is associated with the current session.

### **Return Value**

Returns an array of the names of the controls to hide in the UI for the specified profile.

Returns an empty array if there are no customization settings stored for the supplied profile ID and UI key.

## Remarks

Only users logged in as administrators (ADMIN) can call this method.

## AccpacSession.GetProfiles

Retrieves a list of all UI customization profiles set up in ACCPAC's Administrative Services.

Sub GetProfiles(
ProfileIDs() As String,
ProfileDescs() As String)

## **Parameters**

**ProfileIDs** 

[out] returns an array of profile IDs

**ProfileDescs** 

[out] returns an array of profile descriptions - one item for each item in the profile ID array in the same respective position

## **Remarks**

Only users logged in as administrators (ADMIN) can call this method.

## AccpacSession.GetSignonInfo

Returns the signon information of the current session in one call.

Sub GetSignonInfo(
UserID As String,
CompanyID As String,
CompanyName As String)

## **Parameters**

#### UserID

[out] returns the ID of the currently logged in user

## CompanyID

[out] returns the ID of the company associated with the current open session

## **CompanyName**

[out] returns the database name of the company (as set in ACCPAC Database Setup) associated with the current open session

## **AccpacSession.GetUserCustomizations**

Retrieves UI customization settings for the current logged in user on the UI specified by *UIKey*.

Function GetUserCustomizations(UIKey As String) As String()

### **Parameters**

**UIKey** 

[in] the unique key (identifier) for the UI This UI must be part of the application whose <u>ApplD</u> is associated with the current session.

### **Return Value**

Returns an array of Strings containing the names of the controls to hide in the UI for the current logged in user.

Returns an empty array if there are no customization settings defined for the current user.

## AccpacSession.GetVersion

Retrieves the version information of the ACCPAC System Manager program to which this session is connected.

```
Sub GetVersion(
majorVersion As Long,
minorVersion As Long,
Build As Long,
Revision As Long)
```

### **Parameters**

```
majorVersion
  [out] returns ACCPAC's major version number

minorVersion
  [out] returns ACCPAC's minor version number

Build
  [out] returns ACCPAC's build number

Revision
  [out] returns ACCPAC's revision number
```

# AccpacSession.GetVersion2

Retrieves the version information of the ACCPAC System Manager program to which this session is connected.

Sub GetVersion2() As String

### **Remarks**

Script languages, such as Javascript, which do not support "out" parameters should use this variation of the <u>GetVersion</u> method.

## AccpacSession.Init

Initializes a session with ACCPAC System Manager and opens the session if a valid object handle is supplied.

A Session object must first be initialized before any other object methods can be called.

### Sub Init(

ObjectHandle As String, AppID As String, ProgramName As String, AppVersion As String)

#### **Parameters**

### **ObjectHandle**

[in] the object handle, or "", if the object handle is not available The object handle is passed by other System Manager routines and allows sessions to be opened without supplying a username and password. It is usually available if the application is launched by the ACCPAC desktop, or by another ACCPAC application.

## **AppID**

[in] the two-letter application ID of the application object (also known as a registered module prefix, and has the form XX, such as "AR") If the caller is not a registered ACCPAC module, use "XZ" which is reserved for macros and applications that do not have an ACCPAC registered module prefix.

## **ProgramName**

[in] the Roto ID of the application

The program name should be of the form XXNNNN where XX is the application ID and NNNN is a number. If the caller is not a registered ACCPAC module, NNNN can be any number, such as "XZ1000".

**AppVersion** 

[in] the 3-character application version of the form NNX, such as "51A"

## Remarks

If a valid object handle is supplied (obtained from a previous call to <a href="CreateObjectHandle">CreateObjectHandle</a> or when the application is launched by the ACCPAC desktop), the session is initialized and immediately opened. If an object handle is not available, an empty string ("") should be passed in and the session will be initialized but not opened. The application must then call <a href="Open">Open</a> to open the session.

# AccpacSession.IsWinUserLoggedOn

Determines if the specified Windows user is the user that is currently logged on to the session.

Sub IsWinUserLoggedOn(
Domain As String,
WinUserID As String,
pVal As Boolean)

### **Parameters**

Domain

[in] the Windows domain the user is logged on to

WinUserID

[in] the Windows user ID on the supplied domain

pVal

[out] returns TRUE if the current ACCPAC session user and the Windows user are the same; otherwise, returns FALSE

## AccpacSession.LicenseStatus

Determines whether a valid license is installed for an application.

Function LicenseStatus( AppID As String, AppVersion As String) As <u>tagLicenseStatus</u>

### **Parameters**

```
AppID
[in] the application ID

AppVersion
[in] the application version
```

### **Return Value**

Returns the status of the license for the supplied application ID and version.

## **Remarks**

AppVersion can be passed an empty string (""), in which case the current System Manager version will be assumed.

# AccpacSession.MacroPause

Pauses macro recording for the current session.

Sub macroPause()

## Remarks

To resume macro recording, call <u>MacroResume</u>.

## AccpacSession.MacroRecordObject

Sets up automatic macro recording of the specified view object.

Sub macroRecordObject( hWnd As Long, objectName As String)

### **Parameters**

hWnd
[in] the program's main window handleobjectName
[in] the name of the object in the initialization file for this view

#### Remarks

An application must call this method for automatic macro recording to take place. All views associated with *objectName* will be included in macro recording. This method must be called before views are opened; that is, before DataSource controls are activated.

Macro recording will not work for remote sessions; that is, when IsRemote is TRUE.

Macro recording for the application will end when the passed-in handle (hWnd) is destroyed.

# AccpacSession.MacroResume

Resumes macro recording for the current session.

Sub macroResume()

## **Remarks**

Call MacroResume after pausing macro recording with a <u>MacroPause</u> call.

## AccpacSession.Open

Opens a session to System Manager with the specified user ID and password, and with the database ID of the company to connect to (later on).

### Sub Open(

UserIdentifier As String,

Password As String,

DataBase As String,

SessionDate As Date,

Flags As Long,

Reserved As String)

#### **Parameters**

#### Userldentifier

[in] a valid ACCPAC user ID

#### Password

[in] the user's password

If the specified company database has <u>security</u> enabled, the password is validated. Otherwise, the password is ignored.

#### Database

[in] the database ID (org ID) of the company to open

#### SessionDate

[in] the date to be used for the session

## Flags

[in] reserved - must be 0

#### Reserved

[in] reserved - must be ""

## Remarks

Although a database ID has to be specified to open a session, no database connection is established until the application calls <a href="OpenDBLink">OpenDBLink</a> to create a database link.

The session must be opened before other methods in the object become available.

## AccpacSession.OpenDBLink

Creates a database link to the specified database. A database link can be created on the company database, or the system database that the signed on company is attached to.

```
Function OpenDBLink(
LinkType As tagDBLinkTypeEnum,
Flags As tagDBLinkFlagsEnum) As AccpacDBLink
```

#### **Parameters**

LinkType
[in] specifies the type of database link to create

Flags

[in] specifies the read/write access mode of the database link

### **Return Value**

Returns an AccpacDBLink object, which represents the database link.

## Remarks

The AccpacSession object represents a session to a company that is specified when the session is opened. OpenDBLink creates a database connection to that company, if the selected link type is a company database. If a system database is selected, this method creates a database connection to the system database that is associated with the session's company.

If OpenDBLink is called multiple times to create multiple <u>AccpacDBLink</u> objects, multiple database connections are created and each AccpacDBLink object represents a separate database connection.

## AccpacSession.OpenWin

Establishes an ACCPAC session using Windows authentication.

Sub OpenWin(
Domain As String,
UserIdentifier As String,
Password As String,
DataBase As String,
SessionDate As Date,
Flags As Long,
Reserved As String)

#### **Parameters**

#### Domain

[in] the name of the Windows domain that the user belongs to

#### UserIdentifier

[in] uses the Windows logon id on the specified domain as the ACCPAC user ID

#### Password

[in] the password of the user If the specified company database has <u>security</u> enabled, the password is validated. Otherwise, the password is ignored.

#### Database

[in] the company Database ID (org ID) to use for this session

#### SessionDate

[in] the date to use for this session

## Flags

[in] reserved - must be 0

Reserved [in] reserved - must be ""

### Remarks

Although a Database ID has to be specified to open a session, no database connection is established until the application calls <a href="OpenDBLink">OpenDBLink</a> to create a database link.

The session must be opened before other methods in the object become available.

## AccpacSession.RemoteConnect

Connects the session to a remote ACCPAC server using the supplied authentication credentials.

This method must be called before Init.

Sub RemoteConnect(
ServerName As String,
UserID As String,
Password As String,
Domain As String)

#### **Parameters**

#### ServerName

[in] the name of the ACCPAC server

#### UserID

[in] the username to use to log on to the server
If enhanced security is configured on the server, the ACCPAC
remoting architecture uses NT authentication to verify access rights
of the supplied user. The user must be a user recognized on the
server. This parameter is ignored if enhanced security is not enabled
on the server.

#### Password

[in] the user's password If enhanced security is enabled on the server, the password is validated. Otherwise, the password is ignored.

#### Domain

[in] the Windows domain the supplied user name belongs to, or the server's machine name if the user is a local user on the server This parameter is ignored if enhanced security is turned off on the server.

## AccpacSession.ReportSelect

Creates and returns an AccpacReport object that represents an ACCPAC report.

Function ReportSelect(
bstrReportName As String,
menuID As String,
ProgramID As String) As AccpacReport

### **Parameters**

### *bstrReportName*

[in] the name of the report (the report name must be defined in the application's report INI file)

#### menuID

[in] the menu ID used to identify the property
Menu IDs are assigned by the system and applications do not have
control over the menu ID. If a specific menu ID is not known, an
application should pass in the object ID or an empty string (""). In
these cases, the Session object will use the system-assigned menu
ID.

## **ProgramID**

[in] the Roto ID of the application

The Report object uses this together with the menu ID to locate default report settings saved in the user's properties file.

## **Return Value**

Returns the newly created <u>AccpacReport</u> object, if the specified report name is a valid report defined in the application's report INI file.

## AccpacSession.RscGetString

Retrieves a string from the application's server-side string resource file.

```
Function RscGetString(
AppID As String,
strID As Long) As String
```

### **Parameters**

```
AppID
[in] the two-letter application ID

strID
[in] the resource ID of the string in the application resource file
```

#### **Return Value**

Returns the resource string that matches the specified resource ID. Returns an empty string ("") if the resource ID could not be found in the resource file, or if the specified application ID is not an activated application on the current company.

## Remarks

This function only requires the application ID. The application version is determined automatically, and is the activated version of the specified application on the current company.

# AccpacSession.SaveProfileCustomization

Saves UI customization settings for the specified profile IDs on the UI specified by *UIKey*.

Sub SaveProfileCustomizations(
ProfileIDs() As String,
UIKey As String,
HiddenControls() As String)

#### **Parameters**

#### **ProfileIDs**

[in] an array of profile ID strings

### **UIKey**

[in] the unique key (identifier) for the UI
This UI must be part of the application whose <u>ApplD</u> is associated with the current session.

#### **HiddenControls**

[in] an array containing the names of the controls to hide in the UI for the profiles in *ProfileIDs* 

### Remarks

Only users logged in as administrators can call this method.

Multiple profiles IDs can be specified in the array and the settings are saved to all of those profiles. HiddenControls is an array of names of controls to hide on the UI when customization based on any of those profiles is applied on the UI.

This method is used by the Customize menu option.

This method can only be called when the current user is an administrator

(ADMIN).

## AccpacSession.SetLegacyReturnCode

Sets the return (status) code given to legacy ACCPAC applications when the current program closes.

This return code will be passed back to the program that launched the current application through direct Roto calls, and expects a return code in the Roto's UserArea structure. For more information, refer to the ACCPAC Advantage Series SDK Programming Guide.

Sub SetLegacyReturnCode(ReturnCode As Integer)

#### **Parameters**

ReturnCode [in] return code to pass to legacy applications

#### Remarks

This method is mainly used by activation UIs.

## AccpacSession.TransferFiles

Transfers files between the server and the client.

Function TransferFiles(
SourceLocation As tagFileLocation,
SourcePath As String,
DestLocation As tagFileLocation,
DestPath As String) As Long

#### **Parameters**

#### SourceLocation

[in] specifies whether the source file is on the client machine or on the ACCPAC server

#### SourcePath

[in] the path of the source file; the path must be local if the file is on the client machine; otherwise, the path must be relative to the company subdirectory on the ACCPAC server

#### DestLocation

[out] returns whether to transfer the source file to the client machine or to the ACCPAC server

#### DestPath

[out] returns the path of the transferred file; the path must be local if the file is transferred to the client machine; otherwise, the path must be relative to the company subdirectory on the ACCPAC server

### **Return Value**

Returns the result code of the operation.

# AccpacSession.AppID

Returns the application ID of the current session. This is the application ID used to initialize the session. The application ID is the 2-letter application prefix that identifies the application module, such as AR, OE, and so on.

Property AppID as String (read-only)

## **Remarks**

The application ID is set when the object is initialized by calling the <u>Init</u> method. This property returns an empty string ("") if the session is not open.

# **AccpacSession.Application**

Returns the current application object.

Property Application As <u>AccpacSession</u> (*read-only*)

# AccpacSession.AppVersion

Returns the version of the current application. This is the version used to initialize the session. The application version is a 3-character string that identifies the application version, such as "50A".

Property AppVersion as String (read-only)

### **Remarks**

The application version is set when the object is initialized by calling <u>Init</u>. This property returns an empty string ("") if the session is not open.

## AccpacSession.Codebase

Returns the codebase of the current ACCPAC installation. The codebase is the base path where the ACCPAC application files can be found.

Property Codebase as String (read-only)

### **Remarks**

The format of the codebase returned depends on whether the current session is remote or local.

- If the session is remote, the codebase is the URL to the location where the application CAB files can be downloaded.
- If the session is local, the codebase is the file path to the local directory where ACCPAC is installed.

# AccpacSession.CompanyID

Returns the database ID of the company associated with the current open session.

Property CompanyID As String (read-only)

## **Remarks**

This property is only available after the session is opened. It returns an empty string ("") if the session is not open.

## AccpacSession.CompanyName

Returns the database name of the company associated with the current open session.

Property CompanyName as String (read-only)

### **Remarks**

This property is only available after the session is opened. It returns an empty string ("") if the session is not open.

The company name returned is the name of the database set up in ACCPAC Database Setup (for example, "SAMLTD"). It is not the name defined in the Company Profile UI of ACCPAC Common Services.

# **AccpacSession.DatabaseSeries**

Returns the database series of the current ACCPAC installation.

Property DatabaseSeries As <u>tagDatabaseSeries</u> (*read-only*)

## AccpacSession.EnforceAppVersion

Returns/sets whether <u>OpenDBLink</u> calls enforce a match of (the session's) AppVersion and the activated application version in the company database.

Property EnforceAppVersion As Boolean

#### Remarks

The property is TRUE by default. Activation UIs set this property to FALSE.

When this property is TRUE, the AccpacSession object checks the application version (supplied when calling <u>Init</u>) against the activated version of the application in the database. This check is performed when opening database links (when calling <u>OpenDBLink</u>), and the call will return an error if the versions do not match.

Setting this property to FALSE is useful for applications that specifically need to open databases with a different version of activated data. In this case, the application must examine the <a href="AccpacActiveApp">AccpacActiveApp</a> collection to determine the version of activated data and work with the corresponding database table structures.

## **AccpacSession.Errors**

Returns the <u>AccpacErrors</u> object which contains a list of the errors stored in the session.

Property Errors As <u>AccpacErrors</u> (*read-only*)

## **Remarks**

The <u>AccpacErrors</u> object stores errors generated by all objects that originate from the current session. For example, all AccpacView objects instantiated from this session store error messages in the session object's errors collection.

This property is available only after a session is opened. It returns Null if the session is not open.

## AccpacSession.HelpPath

Returns the local path to the ACCPAC help files' directory.

Property HelpPath As String

### **Remarks**

Even though the property itself is read-write, it should be treated as *read-only* by UI programmers.

If the session is remote, the path goes to the **local** directory where the help files are downloaded and accessed. This directory is usually the current user's Temp directory. If the session isn't remote, this property returns the directory where the help files are installed.

## AccpacSession.HelpURL

Returns the path or URL to the ACCPAC help files' directory.

Property HelpURL As String (read-only)

#### Remarks

Unless a UI has a special reason to download a help file, it should use the <u>HelpPath</u> property (along with the file's name) instead of the HelpURL property to locate its application's help file. This is because the (application-wide) help file is downloaded automatically when the UI is started, so there's no need to look for a remote path.

If the session isn't remote, HelpURL returns a DOS path to the directory where the help files are installed. Unlike <u>HelpPath</u>, if the session is remote, HelpURL will return the URL to the (remote) location where the help files can be downloaded. A UI may need to use HelpURL if it needs to download a help file from another application.

This property is only available after the session is opened. It returns an empty string ("") if the session is not open.

# **AccpacSession.IsOpened**

Returns whether the current session is open. An open session represents an authenticated session with ACCPAC System Manager.

Property IsOpened As Boolean (read-only)

## **Remarks**

A session is opened either by calling <u>Open</u> with the correct username (user ID) and password and logging into a valid company, or by calling <u>Init</u> with a valid object handle assigned by System Manager.

# AccpacSession.IsRemote

Returns whether the current session is accessing a remote or out-ofprocess ACCPAC server.

Property IsRemote As Boolean (read-only)

## **Remarks**

A session is a remote when the <u>AccpacSession</u> object is created on a remote machine or out-of-process (with regard to the application) on the local machine. An application accesses an out-of-process ACCPAC server whenever <u>RemoteConnect</u> is called to connect to a server, even if the application is running on the same machine as the web-deployed ACCPAC server.

To determine whether ACCPAC is running on the local machine (regardless of the session being in-process or out-of-process), call <a href="IsserverMachine">IsserverMachine</a> instead.

This property is always FALSE if web-deployment was not set up and the ACCPAC programs are installed locally.

# AccpacSession.IsServerMachine

Returns whether the application is running on the same machine as the web-deployed ACCPAC server (<u>IsRemote</u> is TRUE).

Property IsServerMachine As Boolean (read-only)

## Remarks

If IsRemote is FALSE then the session is running in-process with respect to the application. In this case the IsServerMachine property returns TRUE, since an in-process session is always on the local machine.

# **AccpacSession.Organizations**

Returns an <u>AccpacOrganizations</u> collection object, which is a collection of all the organizations (system and company databases) set up in the current ACCPAC installation.

Property Organizations As <u>AccpacOrganizations</u> (*read-only*)

## **Remarks**

The Organizations object is available only after the session is initialized. If the session is not initialized, this property returns Null.

# **AccpacSession.Parent**

Returns the parent object.

Property Parent As AccpacSession (*read-only*)

# **AccpacSession.ProductSeries**

Returns the product series of the current ACCPAC installation (for example, whether the installation is Enterprise Edition, Corporate Edition, and so forth).

Property ProductSeries As <u>tagProductSeries</u> (read-only)

# **Remarks**

This property is valid only after a session has been opened.

# AccpacSession.ProgramName

Returns the program name (6-character Roto ID) of the current application. This is the program name that was passed into the <u>Init</u> method to initialize the session.

Property ProgramName As String (read-only)

## **Remarks**

The program name is set when the object is initialized by calling <u>Init</u>. This property is only available after the session is opened. It returns an empty string ("") if the session is not open.

A Roto ID consists of the 2-letter application ID followed by 4 digits.

# AccpacSession.ProgramsPathOnServer

Returns the DOS path to the ACCPAC base directory on the server.

Property ProgramsPathOnServer As String (read-only)

#### Remarks

The ACCPAC base directory is the directory in which the ACCPAC system and program files are installed on the server.

This property returns the result of utlGetPlusPath(). When ACCPAC is running locally, the path points to a directory on the user's computer. When running remotely, the path is a "local" path on the server.

# AccpacSession.RightToLeftLanguage

Returns whether the current user language is a right to left language.

Property RightToLeftLanguage As Boolean (read-only)

# AccpacSession.ServerName

Returns the name of the server of a remote session.

Property ServerName As String (read-only)

#### Remarks

Returns an empty string ("") if ACCPAC is run locally (<u>IsRemote</u> is FALSE).

The return value may be:

- the Windows machine name of the ACCPAC Server deployed on an Intranet, such as ACCPACSERVER
- the Internet host name of the ACCPAC Server, such as accpacserver.yourcompany.com
- the Internet IP address of the ACCPAC Server, such as 24.113.14.47 (for .NET Remoting only)

# AccpacSession.SessionDate

Returns the session date used to open the session.

Property SessionDate As Date (read-only)

## **Remarks**

The session date is set when the session is opened, and isn't necessarily the same as the (Windows) system date.

The session date is passed to the session's <u>Open</u> method. If the session is opened with an object handle through the <u>Init</u> method, the current session inherits the session date of the session which created the object handle.

This property returns the system date if the session was not opened.

# AccpacSession.SharedDataPathOnServer

Returns the DOS path to ACCPAC shared data on the server.

Property SharedDataPathOnServer As String (read-only)

#### Remarks

The shared data directory is the parent directory (on the server, if running remotely) of the SITE, USER, and COMPANY directories.

This property returns the result of utlGetSitePath(). When ACCPAC is running locally, the path points to a directory on the user's computer. When running remotely, the path is a "local" path on the server.

# AccpacSession.UserID

Returns the ID (username) of the currently logged in user. It is the user ID used to open the session.

Property UserID As String (read-only)

## Remarks

The user ID is passed to the session's <u>Open</u> method. If the session is opened with a valid object handle through the <u>Init</u> method, the session inherits the user ID of the session which created the object handle.

It returns an empty string ("") if the session is not open.

# AccpacSession.UserLanguage

Returns the 3-letter language code which represents the current user's language preference.

Property UserLanguage As String (read-only)

## **Remarks**

The user's language preference is set by ADMIN in the Users UI of ACCPAC Administrative Services.

Currently supported language codes include:

Language code	Language
CHN	Chinese
ENG	English
ESN	Spanish
FRA	French

This property returns an empty string ("") if the session is not open.

# **AccpacSession Object**

The AccpacSession object represents an authenticated session with ACCPAC System Manager. It also serves as an entry point to other facilities of the <u>ACCPAC COM API</u>. All other objects in the ACCPAC COM API are created either directly or indirectly by the AccpacSession object.

#### Remarks

The Session object is the only object in the ACCPAC COM API that can be directly created by application programs. An application should create a Session object, initialize and open the session before other facilities in the API become available.

A session can be opened through either of these methods:

- Initializing the session with a valid object handle, by calling <u>Init</u>. The
  object handle is usually supplied by System Manager, or if the
  application is launched by another application, the parent
  application. If a valid object handle is supplied during initialization,
  the session is automatically opened as well and does not require an
  explicit call to Open.
- Opening the session by calling <u>Open</u>, and supplying the correct ACCPAC user name and password. Before calling Open, a session object must first be initialized by calling Init, with "" passed in as the object handle.

# **Example**

The following Visual Basic code illustrates how to create and open the AccpacSession object:

Dim Session As AccpacCOMAPI.AccpacSession

Set Session = CreateObject("ACCPAC.Session")
Session.Open "ADMIN", "ADMIN", "SAMLTD", Date, 0, ""

# AccpacView.BlkGet

Performs a block get of field values of the specified array of field IDs, and returns an array of field values.

Sub BlkGet( FieldIDs As Variant, pValues As Variant)

### **Parameters**

#### **FieldIDs**

[in] an array of 1-based field IDs (array of Longs), which represents the fields whose values should be retrieved

#### *pValues*

[out] returns an array of field values associated with the IDs in *FieldIDs* 

### **Remarks**

This method can be used in place of several BlkGet() calls to minimize the number of view calls.

# **Example**

The following example reads the BANK, NAME, and MULTICUR fields of the view BK0001 and stores them in the variable "Values".

Dim BKACCT1header As AccpacCOMAPI.AccpacView mDBLinkCmpRW.OpenView "BK0001", BKACCT1header Dim Values As Variant BKACCT1header.BlkGet Array(1, 2, 17), Values

# AccpacView.BlkGetFields

Performs a block get of field objects of the specified array of fields IDs, and returns an array of view field objects.

Sub BlkGetFields( FieldIDs As Variant, pValues as Variant)

## **Parameters**

**FieldIDs** 

[in] an array of IDs of the fields from which to retrieve the view field objects

*pValues* 

[out] returns an array of AccpacViewField objects

# AccpacView.BlkPut

Changes the field values of multiple fields.

Sub BlkPut( FieldIDs As Variant, pValues As Variant, Verify As Boolean)

### **Parameters**

#### **FieldIDs**

[in] an array of 1-based field IDs representing the fields whose values should be changed

### pValues

[in] an array of values to assign to the fields whose IDs are specified in *FieldIDs* 

## Verify

[in] whether the values are checked for integrity prior to being stored in the fields

# **Example**

The following example puts the array values within the variable "Values" into the BANK, NAME, and MULTICUR fields of view BK0001, without any verification.

Dim BKACCT1header As AccpacCOMAPI.AccpacView mDBLinkCmpRW.OpenView "BK0001", BKACCT1header Dim Values As Variant Values = Array("BANKONE", "First City Bank", True) BKACCT1header.BlkPut Array(1, 2, 17), Values, False

# AccpacView.Browse

Starts a query so that the caller can scan through a stream of logical records. The routine sets the record filtering criteria and direction for subsequent Read and Fetch calls. The starting key is specified by the current contents of the logical record.

```
Sub Browse(
Filter As String,
Ascending As Boolean)
```

#### **Parameters**

Filter

[in] the filter identifier that restricts the logical records that will be retrieved

Ascending

[in] whether subsequent records will be fetched in ascending (TRUE) or descending (FALSE) order

## Remarks

Calling Browse does not change the ordering of records being fetched from the view. To control the ordering, use the <u>Order</u> property to select a key.

# **Example**

The following example uses a filter so that the program will start its scan at the record whose bank code is equal to or greater than "SEATAC". Since Ascending is TRUE, subsequent records will be fetched in ascending order.

Dim BKACCT1header As AccpacCOMAPI.AccpacView

mDBLinkCmpRW.OpenView "BK0001", BKACCT1header BKACCT1header.Browse "BANK>=""SEATAC"", True

# AccpacView.Cancel

Cancels any unsaved changes in the view's revision list.

Sub Cancel()

# Remarks

Cancel calls are used in conjunction with Post calls.

# AccpacView.CancelEx

Cancels any unsaved changes in the view's revision list. This method is a network optimized version of <u>Cancel</u>. It returns the field values of the requested fields.

Sub CancelEx( FieldIDs As Variant, Values As Variant)

#### **Parameters**

#### *FieldIDs*

[in] an array of field IDs which represent the fields from which the values should be returned if a record is successfully retrieved

#### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

## **Remarks**

This method first calls <u>Cancel</u>, and then calls <u>BlkGet</u> to retrieve the field values.

# AccpacView.Clone

Creates the current view object.

Function Clone() As <a href="AccpacView">AccpacView</a>

### **Return Value**

Returns the cloned **AccpacView** object.

### Remarks

Cloning an AccpacView object creates a new AccpacView object with the same properties set for the current view. Internally, a new view instance is created. However, the current composition is not inherited by the cloned view. So, the cloned view is not composed to any composite views and the caller is responsible for calling <a href="Compose">Compose</a> to properly set up composition.

# AccpacView.Close

Closes an open view.

Sub Close()

# AccpacView.Compose

Links a set of views so that each view within a group can access each other without having to repeatedly open new copies of the view.

Sub Compose(ViewOrArrayOfViews)

### **Parameters**

*ViewOrArrayOfViews* 

[in] an array of <u>AccpacView</u> objects that represent the composite views

### **Remarks**

Each view predefines an ordered list of composite views it expects to compose. When calling Compose, the order of the array of View objects must match the order defined for the view. If composition of a certain subview is not required, the corresponding item in the array should be set to Null. Use the <a href="CompositeNames">CompositeNames</a> property to determine the list and order of composite views that the current view expects.

# **Example**

The following example composes the header view (CS0005) with the detail view (CS0006).

Dim CSRATES1header As AccpacCOMAPI.AccpacView mDBLinkSysRW.OpenView "CS0005", CSRATES1header Dim CSRATES1detail As AccpacCOMAPI.AccpacView mDBLinkSysRW.OpenView "CS0006", CSRATES1detail CSRATES1header.Compose Array(CSRATES1detail)

# AccpacView.Delete

Deletes the current, logical record from the database.

Sub Delete()

# AccpacView.Fetch

Retrieves the next logical record according to the direction and filter set in the last Browse call.

Function Fetch() As Boolean

#### **Return Value**

Returns whether a subsequent record exists in the view and is retrieved. If the end of the table is reached, FALSE is returned.

#### Remarks

To lock the record after retrieval, use FetchLock instead of Fetch.

# **Example**

The following example starts scanning at the record whose bank code is equal to or greater than "SEATAC"; then fetches all subsequent records until the end of the table is reached, or an error occurs. By specifying TRUE as the Ascending parameter in the Browse method, records will be fetched in ascending order.

Dim BKACCT1header As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0001", BKACCT1header
Dim Status as Boolean
BKACCT1header.Browse "BANK>=""SEATAC""", True
Status = BKACCT1header.Fetch
Do While Status = True
Status = BKACCT1header.Fetch
Loop

# AccpacView.FetchEx

Retrieves the next record in the view according to the primary key field values of the current record, as well as the current filter and direction set by a previous call to <u>Browse</u>. This method is a network optimized version of <u>Fetch</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function FetchEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

#### **Parameters**

#### **PutFieldIDs**

[in] an array of field IDs which represent the fields from which the values should first be modified before performing record fetching; the new values are specified in the PutFieldValues parameter

#### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as their corresponding field IDs specified in the PutFieldIDs parameter

#### **FieldIDs**

[in] an array of field IDs representing the fields from which the values should be returned if a record is successfully retrieved

#### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

### RecNum

[out] returns the record number if a record is retrieved

### **Return Value**

Returns whether a subsequent record exists in the view and is retrieved. The return value is the same as Fetch.

# **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Fetch</u>, and then finally <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.FetchLock

Retrieves and locks the next record in the view according to the primary key field values of the current record, as well as the current filter and direction set by a previous call to <a href="Browse">Browse</a>.

Function FetchLock() As Boolean

## **Return Value**

Returns whether a subsequent record exists in the view and is retrieved.

# AccpacView.FetchLockEx

Retrieves and locks the next record in the view according to the primary key field values of the current record, as well as the current filter and direction set by a previous call to <a href="Browse">Browse</a>. This method is a network optimized version of <a href="FetchLock">FetchLock</a>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function FetchLockEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

#### **PutFieldIDs**

[in] an array of field IDs representing the fields from which the values should first be modified before performing record fetching; the new values are specified in the PutFieldValues parameter.

#### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as their corresponding field IDs specified in the PutFieldIDs parameter.

#### **FieldIDs**

[in] an array of field IDs representing the fields from which the values should be returned if a record is successfully retrieved

#### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

#### RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether a subsequent record exists in the view and is retrieved. The return value is the same as <u>FetchLock</u>.

## **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>FetchLock</u>, and then finally <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.FilterDelete

Deletes a set of records from the view that satisfy the supplied filter.

```
Sub FilterDelete(
Filter As String,
strictness As tagFilterStrictnessEnum)
```

## **Parameters**

```
Filter
[in] the filter to be applied

strictness
[in] controls how the deletion operation acts on multiple tables
```

# AccpacView.FilterFetch

Retrieves the next record in the view according to the primary key field values of the current record, as well as the current filter and direction.

Function FilterFetch(lock As Boolean) As Boolean

## **Parameters**

lock

[in] indicates whether the record should be locked after it is retrieved

#### **Return Value**

Returns whether a subsequent record exists in the view and is retrieved.

# AccpacView.FilterSelect

Starts a query and establishes the direction for subsequent calls that fetch records from the view.

```
Sub FilterSelect(
Filter As String,
Ascending As Boolean,
Order As Long,
origin As tagFilterOriginEnum)
```

### **Parameters**

```
Filter
[in] a filter that restricts the records that will be retrieved

Ascending
[in] indicates whether subsequent records will be fetched in ascending (TRUE) or descending (FALSE) order

Order
[in] the key index to use for the view

origin
```

[in] the range of records the filter will affect

# AccpacView.GoBottom

Retrieves the last logical record in the view according to the direction and filter set in the last Browse call.

Function GoBottom() As Boolean

### **Return Value**

Returns whether a record is retrieved. If the table is empty, or if no records match the filter set in the last Browse call, FALSE is returned.

# **Example**

The following example retrieves the last record whose bank code is less than "SEATAC". By specifying TRUE as the Ascending parameter in the Browse method, ascending order is used to determine the last record.

```
Dim BKACCT1header As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0001", BKACCT1header
Dim Status as Boolean
BKACCT1header.Browse "BANK<""SEATAC""", True
Status = BKACCT1header.GoBottom
If Status = False Then
    MsgBox "There are no banks before SEATAC."
    Exit Sub
End If
```

# AccpacView.GoBottomEx

Retrieves the last logical record in the view according to the direction and filter set in the last <u>Browse</u> call. This method is a network optimized version of <u>GoBottom</u>. It returns the field values of the requested fields, as well as the record number.

Function GoBottomEx(
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

## **Parameters**

### **FieldIDs**

[in] an array of field IDs representing the fields from which the values should be returned if a record is successfully retrieved

#### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

### RecNum

[out] returns the record number if a record is retrieved

### **Return Value**

Returns whether a record is retrieved. Since this method always retrieves the last record, the return value is FALSE only if there is no record that satisfies the current filter. The return value is the same as <u>GoBottom</u>.

## **Remarks**

This function first calls <u>GoBottom</u>, and then <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.GoNext

Retrieves the next logical record in the view according to the direction and filter set in the last Browse call.

Function GoNext() As Boolean

### **Return Value**

Returns whether a subsequent record exists and is retrieved. If the end of the table is reached, FALSE is returned.

# **Example**

The following example starts scanning at the record whose bank code is equal to or greater than "SEATAC"; then retrieves all subsequent records until the end of the table is reached, or an error occurs. By specifying TRUE as the Ascending parameter in the Browse method, ascending order is used to determine the next record.

Dim BKACCT1header As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0001", BKACCT1header
Dim Status as Boolean
BKACCT1header.Browse "BANK>=""SEATAC""", True
Status = BKACCT1header.GoNext
Do While Status = True
Status = BKACCT1header.GoNext
Loop

# AccpacView.GoNextEx

Retrieves the next logical record in the view according to the direction and filter set in the last <u>Browse</u> call. This method is a network optimized version of <u>GoNext</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function GoNextEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the PutFieldValues parameter

#### PutFieldValues |

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as their corresponding field IDs specified in the PutFieldIDs parameter.

#### **FieldIDs**

[in] returns an array of field IDs representing the fields for which the values should be returned if a record is successfully retrieved

### **Values**

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

## RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether a subsequent record exists and is retrieved. The return value is the same as <u>GoNext</u>.

# **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>GoNext</u>, and finally <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.GoPrev

Retrieves the previous logical record in the view according to the direction and filter set in the last Browse call.

Function GoPrev() As Boolean

### **Return Value**

Returns whether a previous record exists and is retrieved. If the beginning of the table is reached, FALSE is returned.

# **Example**

The following example starts scanning at the record whose bank code is equal to or less than "SEATAC"; then retrieves all previous records until the beginning of the table is reached, or an error occurs. By specifying TRUE as the Ascending parameter in the Browse method, ascending order is used to determine the previous record.

Dim BKACCT1header As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0001", BKACCT1header
Dim Status as Boolean
BKACCT1header.Browse "BANK<=""SEATAC""", True
Status = BKACCT1header.GoPrev
Do While Status = True
Status = BKACCT1header.GoPrev
Loop

# AccpacView.GoPrevEx

Retrieves the previous logical record in the view according to the direction and filter set in the last <u>Browse</u> call. This method is a network optimized version of <u>GoPrev</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function GoPrevEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

#### PutFieldValues |

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

#### **FieldIDs**

[in] an array of field IDs representing fields for which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

## RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether a previous record exists in the view and is retrieved. The return value is the same as GoPrev.

# **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>GoPrev</u>, and finally <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.GoToBookMark

Locates and retrieves the record identified by the supplied bookmark.

Function GoToBookMark(pBookMark) As Boolean

## **Parameters**

pBookMark
[in] bookmark that identifies the intended record

# **Return Value**

Returns whether the intended record exists in the view and is retrieved.

# AccpacView.GoToBookMarkEx

Locates and retrieves the record identified by the supplied bookmark. This method is a network optimized version of <a href="GoToBookMark">GoToBookMark</a>. It returns the field values of the requested fields, as well as the record number.

Function GoToBookMarkEx(pBookMark As Variant, FieldIDs As Variant, Values As Variant, RecNum As Long) As Boolean

## **Parameters**

## pBookMark

[in] bookmark that identifies the intended record

### **FieldIDs**

[in] an array of field IDs representing fields from which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

### RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether the intended record exists in the view and is retrieved. The return value is the same as GoToBookMark.

This function first calls <u>GoToBookMark</u> (on *pBookMark*), and then calls <u>BlkGet</u> to retrieve the field values.

# AccpacView.GoTop

Retrieves the first logical record in the view according to the direction and filter set in the last Browse call.

Function GoTop() As Boolean

### **Return Value**

Returns whether a record was retrieved. If the table is empty, or if no records match the filter set in the last Browse call, FALSE is returned.

# **Example**

The following example retrieves the first record whose bank code is greater than "SEATAC". By specifying TRUE as the Ascending parameter in the Browse method, ascending order is used to determine the first record.

Dim BKACCT1header As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0001", BKACCT1header
Dim Status as Boolean
BKACCT1header.Browse "BANK>""SEATAC""", True
Status = BKACCT1header.GoTop
If Status = False Then
 MsgBox "There are no banks after SEATAC."
 Exit Sub
End If

# AccpacView.GoTopEx

Retrieves the first logical record in the view according to the direction and filter set in the last <u>Browse</u> call. This method is a network optimized version of <u>GoTop</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function GoTopEx(
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

### **Parameters**

#### **FieldIDs**

[in] an array of field IDs representing fields from which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

#### RecNum

[out] returns the record number if a record is retrieved

### **Return Value**

Returns whether a record is retrieved. Since this method always retrieves the first record, the return value is FALSE only if there is no record that satisfies the current filter. The return value is the same as <u>GoTop</u>.

### Remarks

This function first calls GoTop, and then BlkGet to retrieve the field

values. The record number is also retrieved.

# AccpacView.GoToRecordNumber

Locates and retrieves the record identified by the supplied record number.

Function GoToRecordNumber(RecordNumber As Long) As Boolean

## **Parameters**

RecordNumber

[in] record number that identifies the intended record

## **Return Value**

Returns whether the intended record exists in the view and is retrieved.

# AccpacView.GoToRecordNumber

Locates and retrieves the record identified by the supplied record number. This method is a network optimized version of <a href="Months of EoToRecordNumber">GoToRecordNumber</a>. It returns the field values of the requested fields.

Function GoToRecordNumberEx( RecordNumber As Long, FieldIDs As Variant, Values As Variant) As Boolean

### **Parameters**

#### RecordNumber

[in] record number that identifies the intended record

### **FieldID**

[in] an array of field IDs representing fields for which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

### **Return Value**

Returns whether the intended record exists in the view and is retrieved. The return value is the same as <u>GoToRecordNumber</u>.

### Remarks

This function first calls <u>GoToRecordNumber</u> (on RecordNumber), and then <u>BlkGet</u> to retrieve the field values.

# AccpacView.Init

Blanks, zeroes, or defaults (depending on the field type) the contents of each field in a view.

Sub Init()

# AccpacView.InitEx

Blanks, zeroes, or defaults (depending on the field type) the contents of each field in a view. This method is a network optimized version of <u>Init</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Sub InitEx( FieldIDs As Variant, Values As Variant, RecNum As Long)

### **Parameters**

#### **FieldIDs**

[in] an array of field IDs representing the fields for which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields

### RecNum

[out] returns the record number if a record is retrieved

### Remarks

This method first calls <u>Init</u>, and then <u>BlkGet</u> to retrieve the field values. The record number is also retrieved.

# AccpacView.InitPrimaryKeyFields

Initializes the primary key field values of the current record with the primary key field values of the record specified by the supplied bookmark.

Sub InitPrimaryKeyFields(Bookmark)

### **Parameters**

### Bookmark

[in] bookmark of the record for which the primary key field values should be retrieved

## **Remarks**

This method should be used before inserting records to views that use sequenced revision lists.

# AccpacView.Insert

Creates a new record in the database from the contents of the current logical record.

Sub Insert()

## **Remarks**

The key must not currently exist in the database. The integrity of the record is automatically checked, and, if the integrity check fails, the record is not added. Inserting a record in the database causes the key being used to be reset to the primary key, positioned on the record inserted. If the defined key is not the primary one, the position must be re-established, so that <u>Fetch</u> will operate properly.

# AccpacView.InsertEx

Creates a new record in the database from the contents of the current logical record. This method is a network optimized version of <u>Insert</u>. It allows field values to be set to the view before the operation, and also returns field values of the requested fields, as well as the record number.

Sub InsertEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
GetFieldIDs As Variant,
GetFieldValues As Variant,
RecNum As Long)

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

#### **GetFieldIDs**

[out] returns an array of field IDs representing the fields for which the values should be returned if a record is successfully retrieved

### **GetFieldValues**

[out] returns an array of field values for the requested fields

### RecNum

[out] returns the record number if a record is retrieved

# **Remarks**

This method calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Insert</u>, and finally <u>BlkGet</u> (into *GetFieldValues*) to retrieve the field values. The record number is also retrieved.

# AccpacView.macroAppend

Appends a command to the current macro that is being recorded.

Sub macroAppend(cmdInString As String)

# **Parameters**

cmdInString
[in] command to append to the macro

# AccpacView.Post

Commits any pending changes to the database, functionally operating as Insert and Update do, although "Post" is most useful for a view that is maintaining a batch of records, not just a single record.

Sub Post()

# **Remarks**

This method is used primarily on header views where the call has a cascading effect that causes its detail views to update their unposted revisions.

# AccpacView.PostEx

Commits any pending changes to the database. This method is a network optimized version of <u>Post</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields.

Sub PostEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
GetFieldIDs As Variant,
GetFieldValues As Variant)

### **Parameters**

#### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

### **GetFieldIDs**

[out] returns an array of field IDs representing fields for which the values should be returned if a record is successfully retrieved

### **GetFieldValues**

[out] returns an array of field values for the requested fields

This method first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Post</u>, and then <u>BlkGet</u> (into *GetFieldValues*) to retrieve the field values.

# AccpacView.Process

Performs processing on the view. The operations triggered by this method are specific to the view.

Sub Process()

### Remarks

The Process method is used by special types of views to initiate operations such as year-end processing. These operations do not map nicely to the standard view operation such as Insert, Delete, Update and Post.

Immediately after the Process operation, use the <u>LastReturnCode</u> property to get the return code from that operation. A return code of 0 indicates success, a negative return code indicates a warning, and a positive return code indicates an error.

# **Example**

The following example calls Process on the BK0105 view to indicate that the program wants to add/modify checks and deposits, as long as another program isn't doing the same thing at the same time.

Dim BKPROC1 As AccpacCOMAPI.AccpacView
mDBLinkCmpRW.OpenView "BK0105", BKPROC1
Dim BKPROC1Fields As AccpacCOMAPI.AccpacViewFields
Set BKPROC1Fields = BKPROC1.Fields
BKPROC1Fields("PROCESS").PutWithoutVerification "2" 'Bank Process
BKPROC1Fields("OPERATION").PutWithoutVerification "0" 'Process
Status
Dim rc As tagViewReturnCode
BKPROC1.Process
rc = BKPROC1.LastReturnCode

# AccpacView.ProcessEx

Performs view-specific processing. This method is a network optimized version of <u>Process</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields.

Sub ProcessEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
GetFieldIDs As Variant,
GetFieldValues As Variant)

### **Parameters**

#### **PutFieldIDs**

[in] an array of IDs of the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

### **GetFieldIDs**

[out] returns an array of field IDs representing the fields for which the values should be returned if a record is successfully retrieved

#### GetFieldValues

[out] returns an array of field values for the requested fields

### Remarks

This method first calls **BlkPut** (on *PutFieldValues*), and then **Process**, and

finally  ${\underline{\sf BlkGet}}$  (into  ${\it GetFieldValues}$ ) to retrieve the field values.

# AccpacView.Read

Fetches the logical record indexed by the current contents of the view's key fields. This method ignores any filters set in <u>Browse</u> calls. If there are composite views, the caller is usually responsible for reading them in as well, although this varies by view.

Function Read() As Boolean

### **Return Value**

Returns whether the record identified by the current key field values exists and is retrieved. If the record is not found, FALSE is returned.

# **Examples**

The following example opens the rate type view CS0004 and retrieves (reads) the rate type "SP".

```
Dim CSRATETYPS1 As AccpacCOMAPI.AccpacView
mDBLinkSysRW.OpenView "CS0004", CSRATETYPS1
CSRATETYPS1.Fields("RATETYPE").Value = "SP"
Dim Status as Boolean
Status = CSRATETYPS1.Read
If Status = False Then
    Msgbox "The SP record does not exist."
    Exit Sub
End If
```

# AccpacView.ReadEx

Locates and reads the record in the view identified by the current key field values. This method is a network optimized version of Read. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function ReadEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

#### PutFieldValues |

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

#### **FieldIDs**

[in] an array of field IDs representing the fields from which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

## RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether the record identified by the current key field values exists and is retrieved. The return value is the same as Read.

# **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Read</u>, and finally <u>BlkGet</u> (into *Values*) to retrieve the field values. The record number is also retrieved.

# AccpacView.ReadLock

Retrieves and locks the record in the view identified by the current key field values.

Function ReadLock() As Boolean

# **Return Value**

Returns whether the record identified by the current key field values exists and is retrieved.

# AccpacView.ReadLockEx

Fetches the logical record indexed by the current contents of the view's key fields. This method is a network optimized version of <a href="ReadLock">ReadLock</a>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields, as well as the record number.

Function ReadLockEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
FieldIDs As Variant,
Values As Variant,
RecNum As Long) As Boolean

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

#### PutFieldValues |

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

### **FieldIDs**

[in] an array of field IDs representing the fields from which the values should be returned if a record is successfully retrieved

### Values

[out] returns an array of field values for the requested fields; returns Null if no record is retrieved

### RecNum

[out] returns the record number if a record is retrieved

## **Return Value**

Returns whether the record identified by the current key field values exists and is retrieved. The return value is the same as ReadLock.

# **Remarks**

This function first calls <u>BlkPut</u> (on *PutFieldValues*), then <u>ReadLock</u>, and finally <u>BlkGet</u> (into *Values*) to retrieve the field values. Also retrieves the record number. Returns what <u>ReadLock</u> returns.

# AccpacView.RecordClear

Blanks, zeros, or defaults the fields in the view.

Sub RecordClear()

# AccpacView.RecordGenerate

Generates a new unique key, and blanks, zeroes, or defaults the remaining fields in the view.

Sub RecordGenerate(Insert As Boolean)

### **Parameters**

Insert

[in] specifies whether the generated record should be immediately inserted into the view

# AccpacView.ResetRecordNumbers

Resets all record numbers generated by the view. Record numbers will be regenerated the next time a method is called that moves the record pointer.

Sub ResetRecordNumbers()

### Remarks

When <u>UseRecordNumbering</u> is enabled for the view, a record number is generated by the view whenever it fetches a record. An internal table is built that maps the record's bookmarks to the generated record numbers. This method resets the internal mapping table and forces the view to regenerate record numbers when subsequently fetching records.

This method should be used on detail views when changing the header view produces a different set of detail records.

# AccpacView.RevisionCancel

Rolls back any pending changes to the specified revision level.

Sub RevisionCancel(level As Long)

### **Parameters**

level [in] revision level

# AccpacView.RevisionExists

Returns whether the current record exists within the specified revision level.

Function RevisionExists(level As Long) As Boolean

### **Parameters**

*level* [in] revision level

### **Return Value**

Returns whether the current record exists within the specified revision level.

# AccpacView.RevisionPost

Commits any pending changes to the specified revision level.

Sub RevisionPost(level As Long)

### **Parameters**

level [in] revision level

# AccpacView.RevisionUnposted

Returns whether the specified revision level has unposted changes.

Function RevisionUnposted(level As Long) As Boolean

### **Parameters**

level [in] revision level

### **Return Value**

Returns whether the specified revision level has unposted changes.

# AccpacView.TableEmpty

Clears the view's table of all records, using the fastest available method.

Sub TableEmpty()

### **Remarks**

This method calls TableEmpty() on the view. To verify that the routine is available, check the view's TemplateVersion property prior to calling this routine.

# AccpacView.UnLock

Unlocks a record locked by a FetchLock or ReadLock call.

Sub UnLock()

## AccpacView.Update

Writes the contents of the existing logical record back to the database using the current key.

Sub Update()

### Remarks

If the key fields have been altered, use the <u>Insert</u> method instead. If there are composite views, you must update them as necessary.

### **Example**

The following example opens the rate type view CS0004, retrieves (reads) the rate type "SP", and changes the rate type description to "Daily spot rate" before updating the rate type record.

Dim CSRATETYPS1 As AccpacCOMAPI.AccpacView
mDBLinkSysRW.OpenView "CS0004", CSRATETYPS1
CSRATETYPS1.Fields("RATETYPE").Value = "SP"
Dim Status as Boolean
Status = CSRATETYPS1.Read
If Status = False Then
 Msgbox "The SP record does not exist."
 Exit Sub
End If
CSRATETYPS1.Fields("RATEDESC").PutWithoutVerification
("Daily spot rate")
CSRATETYPS1.Update

# AccpacView.UpdateEx

Writes the contents of the existing logical record back to the database using the current key. This method is a network optimized version of <a href="Update">Update</a>. It allows the field values to be set to the view before the operation, and also returns the field values of the requested fields.

Sub UpdateEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
GetFieldIDs As Variant,
GetFieldValues As Variant)

### **Parameters**

### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

#### **GetFieldIDs**

[out] returns an array of field IDs representing the fields for which the values should be returned if a record is successfully retrieved

#### **GetFieldValues**

[out] returns an array of field values for the requested fields

This method first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Update</u>, and finally <u>BlkGet</u> (into *GetFieldValues*) to retrieve the field values.

# AccpacView.Verify

Checks the referential integrity of the view's current record.

Sub Verify()

### **Remarks**

This method will validate the field values of the current record.

Do this check prior to writing the record to the database (before calling <u>Update</u>), particularly if any fields were put into the view without field-level verification.

Immediately after the Verify call, use the <u>LastReturnCode</u> property to get the return code from the referential integrity check. A return code of 0 indicates success, a negative return code indicates a warning, and a positive return code indicates an error.

# AccpacView.VerifyEx

Checks the referential integrity of the view's current record. This method is a network optimized version of <u>Verify</u>. It allows field values to be set to the view before the operation, and also returns the field values of the requested fields.

Sub VerifyEx(
PutFieldIDs As Variant,
PutFieldValues As Variant,
GetFieldIDs As Variant,
GetFieldValues As Variant)

### **Parameters**

#### **PutFieldIDs**

[in] an array of field IDs representing the fields for which the values should first be modified before performing record fetching; the new values are specified in the *PutFieldValues* parameter

### **PutFieldValues**

[in] an array of field values that should first be set to the view before performing record fetching; the values should appear in the array in the same order as the corresponding field IDs specified in the *PutFieldIDs* parameter

#### **GetFieldIDs**

[out] returns an array of field IDs for which the values should be returned if a record is successfully retrieved

#### **GetFieldValues**

[out] returns an array of field values for the requested fields

This method first calls <u>BlkPut</u> (on *PutFieldValues*), and then <u>Verify</u>, and finally <u>BlkGet</u> (into *GetFieldValues*) to retrieve the field values.

# AccpacView.CompositeNames

Returns the <u>AccpacViewCompositeNames</u> collection object that contains the names of the subviews that are composed to the view.

Property CompositeNames As <u>AccpacViewCompositeNames</u> (*read-only*)

### **Remarks**

This property returns an array of Roto IDs of the views that the current view expects to be composed. Applications should open the corresponding view objects using <a href="AccpacDBLink.OpenView">AccpacDBLink.OpenView</a> and call <a href="Compose">Compose</a> to perform the composition.

The order of the view IDs appearing in the array denotes the order of the view objects expected when calling <u>Compose</u>.

# **AccpacView.Description**

Returns the descriptive name of the view.

Property Description As String (read-only)

# AccpacView.Dirty

Returns whether fields in the logical record have been altered.

Property Dirty As Boolean (read-only)

# AccpacView.Exists

Returns whether the current logical record exists in the database.

Property Exists As Boolean (read-only)

# AccpacView.Fields

Returns the <u>AccpacViewFields</u> collection object that represents all the fields in the view.

Property Fields As <u>AccpacViewFields</u> (*read-only*)

# AccpacView.HeaderLinkedKeyFieldCount

Returns/sets the number of key fields linked to the header view.

Property HeaderLinkedKeyFieldCount As Integer

# AccpacView.InstanceNonheritable

Returns whether the composite views implicitly opened by the current view inherit the flags (additional parameters) with which the current view was opened.

Property InstanceNonheritable As Boolean (*read-only*)

# AccpacView.InstancePrefetch

Returns/sets the number of records fetched at a time, if the view was opened in read-only mode.

Property InstancePrefetch As Long (read-only)

# AccpacView.InstanceReadonly

Returns/sets whether the view was opened in read-only mode.

Property InstanceReadonly As Boolean (read-only)

# **AccpacView.InstanceSecurity**

Returns the access rights the current user has on the view.

Property InstanceSecurity As <u>tagViewSecurityEnum</u> (*read-only*)

### **Remarks**

The access rights that a user has on a view can be a combination of the defined <a href="mailto:tagViewSecurityEnum">tagViewSecurityEnum</a> values.

# AccpacView.InstanceUnrevisioned

Returns whether revisioning is suppressed.

Property InstanceUnrevisioned As Boolean (*read-only*)

# AccpacView.InstanceUnvalidated

Returns whether validation is suppressed.

Property InstanceUnvalidated As Boolean (read-only)

# AccpacView.Keys

Returns the <u>AccpacViewKeys</u> collection object that provides access to all keys exposed by the current view.

Property Keys As <u>AccpacViewKeys</u> (*read-only*)

# AccpacView.LastReturnCode

Returns the (ACCPAC API) return code of the last view operation.

Property LastReturnCode As <a href="mailto:tagViewReturnCode">tagViewReturnCode</a> (read-only)

### **Remarks**

This property always returns the view's return code of the last call to a view function.

# AccpacView.Order

Returns/sets the primary view's current key. Changing the selected key affects the order in which records are fetched from the view.

Property Order As Long

### **Remarks**

When a view is first opened, the default selected key is always key 0.

# AccpacView.RecordBookMark

Returns the bookmark of the current record. A bookmark uniquely identifies a record in a view.

Property RecordBookMark As Variant (read-only)

### **Remarks**

The bookmark is an internal representation of the current primary key field values, as well as other information required to uniquely identify a record.

A bookmark is primarily used to reposition the record pointer to the current record by calling <u>GotoBookmark</u>.

# AccpacView.RecordNumber

Returns the record number of the current record.

Property RecordNumber As Long (read-only)

### Remarks

The AccpacView object generates a record number for every record it fetches, if the <u>UseRecordNumbering</u> property is set to TRUE. If <u>UseRecordNumbering</u> is FALSE, this property always returns 0.

Record numbers should only be used on views that use sequenced revisions lists. In all other cases, records are not sequenced in any way, so the AccpacView object cannot generate meaningful record numbers.

# **AccpacView.ReferentialIntegrity**

Returns/sets the current referential integrity flag of the view.

Property ReferentialIntegrity As <u>tagReferentialIntegrityEnum</u>

# **AccpacView.Security**

Returns the access rights the current user has on the view.

Property Security As <u>tagViewSecurityEnum</u> (*read-only*)

## Remarks

The access rights of a user on a view are a combination of the defined <a href="mailto:tagViewSecurityEnum">tagViewSecurityEnum</a> values.

# AccpacView.Session

Returns the AccpacSession object to which the view is indirectly attached.

Property Session As <u>AccpacSession</u> (*read-only*)

# AccpacView.SystemAccess

Returns/sets the system access mode of the view. The view may behave differently for different access modes.

Property SystemAccess As tagSystemAccessEnum

# AccpacView.TemplateDate

Returns the date of the template used by the view.

Property TemplateDate As Date (read-only)

# AccpacView.TemplateVersion

Returns the version of the template used by the view.

Property TemplateVersion As String (read-only)

# AccpacView.Type

Returns the view type.

Property Type As <u>tagRotoViewTypeEnum</u> (*read-only*)

# AccpacView.UnpostedRevisions

Returns whether there are unposted revisions in the view.

Property UnpostedRevisions As Boolean (read-only)

# AccpacView.UsedOpenViewInstance

Returns whether the view was opened with AccpacDBLink.OpenViewInstance.

Property UsedOpenViewInstance As Boolean (read-only)

### AccpacView.UseRecordNumbering

Returns/sets whether the view object generates a record number for each fetched record. Record numbering can be used to access views that use sequenced revision lists.

Property UseRecordNumbering As Boolean

#### **Remarks**

This property should only be set to TRUE for views that use sequenced revision lists. In all other cases, the records are not sequenced in any way, so the AccpacView object cannot generate meaningful record numbers.

# AccpacView.ViewID

Returns the Roto ID of the view.

Property ViewID As String (read-only)

### Remarks

The Roto ID of the view has the form NNXXXX, where NN is the application prefix and XXXX is a number that uniquely identifies the view in the application.

### **AccpacView Object**

The AccpacView object enables you to access and maintain the database through views. The AccpacView object also provides information about a view. Each view is a combination of a logical record (such as an account) or a logical function and its associated action routines (such as posting).

### Remarks

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacDBLink</u> object's <u>OpenView</u> method.

### AccpacViewCompositeNames.Item

Returns the name (Roto ID) of the subview, based on its 0-based index in the collection.

Function Item(Index As Long) As String (*default*)

#### **Parameters**

Index

[in] the 0-based index of the subview in the collection

#### **Return Value**

Returns the Roto ID of a view that the current view composes.

#### Remarks

Because the Item method is the default method for this collection, you do not have to specify "Item." For example, the following syntax forms are interchangeable:

CompositeNames.Item(0)
CompositeNames(0)

### Example

The following example displays the Roto ID of the first subview (BK0002) of the Bank Account view (BK0001) in a message box:

Dim BKACCT1header As AccpacCOMAPI.AccpacView mDBLinkCmpRW.OpenView "BK0001", BKACCT1header MsgBox BKACCT1header.CompositeNames(0)

# AccpacViewCompositeNames.Count

Returns the number of composite views contained in the collection.

Property Count As Long (read-only)

#### **Return Value**

Returns a count of the views that the current view composes.

## **AccpacViewCompositeNames Object**

Contains a collection of subview names.

#### **Remarks**

This object provides access to a collection of Roto IDs that represent the view the current view composes.

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacView</u> object's <u>CompositeNames</u> property.

### AccpacViewField.GetInfo

Retrieves all of the view field's information in one call.

```
Sub GetInfo(
Index As Long,
ID As Long,
Name As String,
Description As String,
Type As tagFieldTypeEnum,
Attribs As tagFieldAttributeEnum,
Size As Integer,
Precision As Integer,
MaxValue As Variant,
MinValue As Variant,
PresentationType As tagFieldPresentationTypeEnum,
PresentationMask As String,
Value As Variant)
```

#### **Parameters**

```
Index
[out] returns the index of the field in the 0-based collection

ID
[out] returns the 1-based field ID, as defined in the view

Name
[out] returns the field name

Description
[out] returns the field description

Type
[out] returns the field type
```

#### **Attribs**

[out] returns the attributes of the current state of the field

#### Size

[out] returns the field size (the number of bytes of storage required by the field)

#### Precision

[out] returns the precision, if the field is of numeric type (*Type* is FLD\_BCD)

#### MaxValue

[out] returns the maximum value of the field, according to the field type

#### **MinValue**

[out] returns the minimum value of the field, according to the field type

#### PresentationType

[out] returns the type of presentation information for the field

#### PresentationMask

[out] returns the field's presentation mask, if the presentation type specifies mask

#### Value

[out] returns the current value stored in the field

### AccpacViewField.PutWithoutVerification

Sets a new value to the field without validation.

Sub PutWithoutVerification(pNewVal As Variant)

#### **Parameters**

pNewVal[in] the new value to be stored in the fieldThe data type of the new value must be compatible with the field type.

#### **Remarks**

This new value is not validated by the view.

### **Example**

The following example stores the value "Monthly average rate" in the RATEDESC field without verifying its integrity, for the rate type "AV".

Dim CSRATETYPS1 As AccpacCOMAPI.AccpacView mDBLinkSysRW.OpenView "CS0004", CSRATETYPS1 Dim CSRATETYPS1Fields As AccpacCOMAPI.AccpacViewFields Set CSRATETYPS1Fields = CSRATETYPS1.Fields CSRATETYPS1Fields("RATETYPE").Value = "AV" CSRATETYPS1Fields("RATEDESC").PutWithoutVerification\_ "Monthly average rate" CSRATETYPS1.Insert

# AccpacViewField.SetToMax

Sets the field value to the maximum value, as defined by the current <u>MaxValue</u> property.

Sub SetToMax()

# AccpacViewField.SetToMin

Sets the field value to the minimum value, as defined by the current <u>MinValue</u> property.

Sub SetToMin()

## AccpacViewField.Attributes

Returns a bitset of status indicators and attributes of the current state of a field.

Property Attributes As <u>tagFieldAttributeEnum</u> (*read-only*)

### **Remarks**

The run time attributes of a field can be a combination of the values defined in the <a href="mailto:tagFieldAttributeEnum">tagFieldAttributeEnum</a> enumeration.

# AccpacViewField.Description

Returns the field description.

Property Description As String (read-only)

### **Remarks**

The description is localized according to the user's language preference.

# AccpacViewField.ID

Returns the 1-based field ID that corresponds to the field index defined in the view.

Property ID As Long (read-only)

### **Remarks**

Field IDs are unique within a view.

## AccpacViewField.Index

Returns the field's index in the view's 0-based <u>AccpacViewFields</u> collection.

Property Index As Long (read-only)

#### **Remarks**

The field's index in the <u>AccpacViewFields</u> collection should not be confused with its ID defined in the view.

## AccpacViewField.MaxValue

Returns the maximum allowable value for the field.

Property Max Value as Variant (read-only)

#### **Remarks**

The maximum value is based solely on the field type, without regard for business logic.

If the field uses a presentation list, the maximum value is the value of the last presentation item in the list.

## AccpacViewField.MinValue

Returns the minimum allowable value for the field.

Property MinValue As Variant (read-only)

#### **Remarks**

The minimum value is based solely on the field type, without regard for business logic.

If the field uses a presentation list, the minimum value is the value of the first presentation item in the list.

# AccpacViewField.Name

Returns the field name.

Property Name As String (read-only)

### **Remarks**

Field names are unique within a view.

# AccpacViewField.Precision

The number of decimal places, if the field type is FLD\_BCD.

Property Precision As Integer (read-only)

## AccpacViewField.PresentationMask

Returns the field's presentation mask, if the presentation type specifies mask.

Property PresentationMask As String (read-only)

### **Remarks**

A presentation mask controls the field's display format. This property returns an empty string ("") if the field does not use a presentation mask.

## AccpacViewField.PresentationStrings

Returns an <u>AccpacViewFieldPresentsStrings</u> object which stores the presentation strings defined for the field.

Property PresentationStrings As <u>AccpacViewFieldPresentsStrings</u> (*read-only*)

#### **Remarks**

Returns Null if the field does not have a presentation list.

## AccpacViewField.PresentationType

The type of presentation information for the field (that is, whether there is a presentation mask or presentation strings for the field).

Property PresentationType As <u>tagFieldPresentationTypeEnum</u> (*read-only*)

## AccpacViewField.Size

Returns the number of bytes of storage required by the field.

Property Size As Integer (read-only)

#### **Remarks**

This property represents the storage space required by the field to store its value in the database. The Size property does not necessarily equal the memory allocated for a field object created programmatically.

# AccpacViewField.Type

Returns the data type of the field.

Property Type As <a href="mailto:tagFieldTypeEnum">tagFieldTypeEnum</a> (read-only)

## AccpacViewField.Value

Returns/sets the contents of the field. If you assign a new value to the field, the value is automatically verified. If the value fails verification, the value is not stored.

Property Value As Variant (default)

### Remarks

This is the default property.

# AccpacViewField.View

Returns the ACCPAC view object to which this field belongs.

Property View As <u>AccpacView</u> (*read-only*)

## **AccpacViewField Object**

Represents a field in an ACCPAC view. This class provides methods and properties to access details of a view field and manipulate field values.

#### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacViewFields</u> collection.

### AccpacViewFieldPresentsStrings.GetAll

Retrieves all of the presentation strings collection's properties in one method call.

Sub GetAll(
Count As Long,
Cookies As Variant,
Strings As Variant,
Values As Variant)

#### **Parameters**

#### Count

[out] returns the number of presentation strings in the collection

#### Cookies

[out] returns an array of cookies corresponding to the presentation strings (array of Longs); the cookies are used for looking up presentation strings

### Strings

[out] returns an array of presentation strings (array of Strings)

#### Values

[out] returns an array of field values corresponding to the presentation strings (array of Variants)

## AccpacViewFieldPresentsStrings.Refresh

Refreshes the presentation strings from the view.

Sub Refresh()

#### **Remarks**

Applications should call this method to refresh the list if the field attributes indicate that the presentation information may change. Refreshing the list ensures that the field retrieves the most current presentation list as of the current view state.

## AccpacViewFieldPresentsStrings.SetField

Sets the current field value to the predefined value of a presentation item, where the presentation item is identified by the supplied cookie.

Sub SetFieldValue(Cookie As Long)

### **Parameters**

Cookie

[in] the cookie used to look up the presentation string and its associated field value

## AccpacViewFieldPresentsStrings.AllPred

Returns an array of predefined presentation strings and their corresponding cookies.

Property AllPredefinedString(pCookies s Variant)As Variant (read-only)

#### **Parameters**

*pCookies* 

[out] returns an array of cookies that correspond to the presentation string list

# AccpacViewFieldPresentsStrings.Count

Returns the number of predefined presentation strings in the collection.

Property Count As Long (read-only)

## AccpacViewFieldPresentsStrings.FieldStr

Returns the predefined presentation string defined for the current field value.

Property FieldString As String (read-only)

### AccpacViewFieldPresentsStrings.Predefin

Returns the predefined display string and cookie of a presentation item according to the item's zero-based index in the list.

```
Property PredefinedString(
Index As Long,
pCookie As Long) As String
(read-only)
```

#### **Parameters**

```
Index
[in] the index of the presentation item in the zero-based list

pCookie
[out] returns the cookie that corresponds to the presentation string
```

## AccpacViewFieldPresentsStrings.Predefin

Returns the predefined value of a presentation item, specified by the cookie associated with the presentation item.

Property PredefinedValue(Cookie As Long) (read-only)

#### **Parameters**

Cookie

[in] the cookie which identifies the presentation item in the list

### **AccpacViewFieldPresentsStrings Object**

Provides access to the presentation string collection that is exposed by a field in an ACCPAC view.

#### Remarks

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacViewField</u> object's <u>PresentationString</u> property.

A presentation string collection is a predefined list of valid values allowed by a field, and their corresponding display strings. The valid values should be treated as internal values and should not be displayed to the user. Users should be presented with the presentation strings which are meaningful to them, and localized to the user's language preference.

### AccpacViewFields.Item

Returns a specific <u>AccpacViewField</u> object in the collection, specified by its 0-based index in the collection or by its field name.

Function Item(pIndex As Variant) As <a href="https://example.com/AccpacViewField">AccpacViewField</a> (default)

#### **Parameters**

```
pIndex
[in] the index of the AccpacViewField object in the 0-based collection, or the field name
```

#### **Return Value**

Returns the <u>AccpacViewField</u> object corresponding to the supplied index.

#### Remarks

Because the Item method is the default method for this collection, you do not have to specify "Item." For example, the following syntax forms are interchangeable:

```
Field.Item(0)
Fields(0)
Fields.Item("name")
Fields("name")
```

Note that the index denotes the position of the field stored in the collection, and does not represent the field ID defined in the view.

#### **Example**

The following example retrieves the NAME field from the Bank Account's fields collection:

Dim BKACCT1header As AccpacCOMAPI.AccpacView mDBLinkCmpRW.OpenView "BK0001", BKACCT1header Dim fldBANK As AccpacCOMAPI.AccpacViewField Set fldBANK = BKACCT1header.Fields("BANK")

## AccpacViewFields.Count

Returns the number of <u>AccpacViewField</u> objects in the collection.

Property Count As Long (read-only)

#### **Return Value**

Returns the number of fields exposed by the view.

## AccpacViewFields.FieldByID

Returns the <u>AccpacViewField</u> object by its 1-based field ID defined in the view.

Property FieldByID(ID As Long) As <u>AccpacViewField</u> (read-only)

#### **Parameters**

ID

[in] the Field ID of the desired field, where the Field ID is the index of the field as defined in the view

#### **Return Value**

Returns the AccpacViewField object of the desired field.

# AccpacViewFields.FieldByIndex

Returns the <u>AccpacViewField</u> object by its 0-based index in the collection.

Property FieldByIndex(Index As Long) As <u>AccpacViewField</u> (read-only)

#### **Parameters**

Index

[in] the index of the AccpacViewField object in the 0-based collection

#### **Return Value**

Returns the AccpacViewField object of the desired field.

## AccpacViewFields.FieldByName

Returns the <u>AccpacViewField</u> object corresponding to the specified field name.

Property FieldByName(Name As String) As <u>AccpacViewField</u> (read-only)

#### **Parameters**

Name

[in] the name of the desired field

#### **Return Value**

Returns the AccpacViewField object of the desired field.

## **AccpacViewFields Object**

Contains a collection of fields (<u>AccpacViewField</u> objects) exposed by a view.

#### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacView</u> object's <u>Fields</u> property.

## AccpacViewKey.Field

Returns the <u>AccpacViewField</u> object by the 0-based index in the collection.

Property Field(Index As Long) As <u>AccpacViewField</u> (read-only, default)

#### **Parameters**

Index

[in] the index of the field in the 0-based collection

# AccpacViewKey.FieldCount

Returns the number of fields in the key.

Property FieldCount As Long (read-only)

# AccpacViewKey.FieldValues

Returns/sets an array of the values of all the fields in the key.

Property FieldValues As Variant

# AccpacViewKey.Name

Returns the name of the key.

Property Name As String (read-only)

## **AccpacViewKey Object**

Provides information about a key defined in an ACCPAC view.

#### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacViewKeys</u> class.

### AccpacViewKeys.Item

Returns a specific AccpacViewKey object in the collection, specified by its 0-based index in the collection or by its key name.

Function Item(Index As Variant) As <a href="https://example.com/AccpacViewKey">AccpacViewKey</a> (default)

#### **Parameters**

Index

[in] the index of the AccpacViewKey object in the 0-based collection, or the key name

#### **Return Value**

Returns the AccpacViewKey object corresponding to the supplied index.

#### Remarks

Because the Item method is the default method for this collection, you do not have to specify "Item." For example, the following syntax forms are interchangeable:

Keys.Item(0) Keys(0) Keys.Item("name") Keys("name")

### **Example**

The following example retrieves the first key from the Bank Account's keys collection:

Dim BKACCT1header As AccpacCOMAPI.AccpacView mDBLinkCmpRW.OpenView "BK0001", BKACCT1header Dim keyBKACCT As AccpacCOMAPI.AccpacViewKey Set keyBKACCT = BKACCT1header.Keys(0)

# AccpacViewKeys.Count

Returns the number of AccpacViewKey objects in the collection.

Property Count As Long (read-only)

#### **Return Value**

Returns the number of keys defined in the view.

# **AccpacViewKeys Object**

This object exposes a collection of <u>AccpacViewKey</u> objects, which represent all keys defined for an ACCPAC view.

#### **Remarks**

An object of this class cannot be created directly by applications. It should be obtained from the <u>AccpacView</u> object's <u>Keys</u> property.