

# Commence Database API

This Guide documents the Commence Database API introduced with Commence 4.1 and the Automation objects that it supports. The Reference section details the properties and methods of each of the objects.

## Reference

[ICommenceDB](#)

[ICommenceCursor](#)

[ICommenceConversation](#)

[ICommenceQueryRowset](#)

[ICommenceAddRowset](#)

[ICommenceEditRowset](#)

[ICommenceDeleteRowset](#)

[Constants](#)

[Developer Notes](#)

# **ICommenceAddRowSet Object**

Represents the set of new items to add to the database

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Properties**

### **Long ColumnCount**

(read-only) Number of columns in this rowset (-1 on error).

### **Long RowCount**

(read-only) Number of rows in this rowset (-1 on error).

## **Methods**

### **Long Commit**

Make row modifications permanent (commit to disk).

### ICommenceCursor **CommitGetCursor**

Make row modifications permanent (commit to disk) and return a cursor

### **Long GetColumnIndex**

Search the column set and return the index of the column with the given label.

### **String GetColumnLabel**

Return the label associated with a particular column.

### **String GetRow**

Returns an entire row's field values.

### **String GetRowValue**

Returns the field value at the given (row,column) in text form.

### **Long ModifyRow**

Modify a field value in the rowset. for the newly added data.

### **Boolean SetShared**

Mark a row to be shared.

### **Boolean GetShared**

Return the row's current shared vs. local status.

# **ICommenceAddRowSet::ColumnCount**

## **Data Type**

Long

## **Description**

(read-only) Number of columns in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **See Also**

[ICommenceAddRowSet](#)

# **ICommenceAddRowSet::RowCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of rows in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **See Also**

[ICommenceAddRowSet](#)

# **ICommenceAddRowSet::GetRowValue**

**String GetRowValue(Long *nRow*, Long *nCol*, Long *nFlags*)**

Returns the field value at the given (row,column) in text form.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns Field value in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

See the Developer Notes for more information about the canonical format.

## **See Also**

[ICommenceAddRowSet](#)

[RowCount](#)

[ColumnCount](#)

[GetRow](#)

# **ICommenceAddRowSet::GetColumnLabel**

**String GetColumnLabel(Long *nCol*, Long *nFlags*)**

Return the label associated with a particular column.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns Column label in text form on success, NULL on error.

## **Parameters**

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will return the labels used with the view.

Specify the CMC\_FLAG\_FIELD\_NAME to force the underlying Commence field name to be returned.

## **See Also**

ICommenceAddRowSet

GetColumnIndex

# **ICommenceAddRowSet::GetColumnIndex**

**Long GetColumnIndex(String *pLabel*, Long *nFlags*)**

Search the column set and return the index of the column with the given label.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns 0-based column index on success, -1 on error.

## **Parameters**

*pLabel*

The column label to map.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will first search the view labels for a possible match. If not found, the field labels will then be searched. Specify the CMC\_FLAG\_FIELD\_NAME to force only the underlying Commence field name to be searched.

## **See Also**

[ICommenceAddRowSet](#)

[GetColumnLabel](#)

# **ICommenceAddRowSet::ModifyRow**

**Long ModifyRow(Long *nRow*, Long *nCol*, String *pBuf*, Long *nFlags*)**

Modify a field value in the rowset.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*pBuf*

New field value in text form.

*nFlags*

Unused at present, must be 0.

## **Comments**

Modifications to the rowset will be reflected by GetRowValue() and GetRow() but changes are not permanent until Commit() is called.

## **See Also**

ICommenceAddRowSet

Commit

CommitGetCursor



# **ICommenceAddRowSet::Commit**

**Long Commit**(Long *nFlags*)

Make row modifications permanent (commit to disk).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nFlags*

Unused at present, must be 0.

## **Comments**

After Commit(), the ICommenceAddRowSet is no Longer valid and should be discarded.

## **See Also**

[ICommenceAddRowSet](#)

[ModifyRow](#)

[CommitGetCursor](#)

# **ICommenceAddRowSet::CommitGetCursor**

**ICommenceCursor CommitGetCursor(Long *nFlags*)**

Make row modifications permanent (commit to disk) and return a cursor for the newly added data.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns ICommenceCursor object on success, NULL on error.

## **Parameters**

*nFlags*

Unused at present, must be 0.

## **Comments**

After Commit(), the ICommenceAddRowSet is no Longer valid and should be discarded.

## **See Also**

[ICommenceAddRowSet](#)

[ModifyRow](#)

[Commit](#)

# **ICommenceAddRowSet::GetRow**

**String GetRow(Long *nRow*, String *pDelim*, Long *nFlags*)**

Returns an entire row's field values.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns Row values in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*pDelim*

Delimiter to use between field values.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

pDelim is used to separate field values. pDelim can be up to 20 chars.

Returned string is EOS terminated. Format is: <col1><delim><col2><delim>...<coln><EOS>

## **See Also**

ICommenceAddRowSet

GetRowValue

# **ICommenceAddRowSet::SetShared**

**Boolean SetShared(Long *nRow*)**

Mark a row to be shared.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **Comments**

By default, all rows are marked local. To create a rowset with all rows marked as shared, call GetAddRowSet with the CMC\_FLAG\_SHARED flag.

## **See Also**

[ICommenceAddRowSet](#)

[ICommenceCursor::GetAddRowSet](#)

# **ICommenceAddRowSet::GetShared**

**Boolean** **GetShared**(Long *nRow*)

Return the row's current shared vs. local status.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/ADD.D

## **Return Value**

Returns TRUE if row is shared, FALSE otherwise.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **See Also**

[ICommenceAddRowSet](#)

# **ICommenceConversation Object**

Represents a DDE conversation.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CONVERSATION.D

## **Methods**

### **Boolean Execute(String pDDECommand)**

Executes the DDE Command.

### **String Request(String pDDECommand)**

Processes the DDE Request.

# **ICommenceConversation::Execute**

**Boolean Execute(String *pDDECommand*)**

Executes the DDE Command.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CONVERSATION.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*pDDECommand*

Text with the DDE command. Syntax is identical to the commands used by the DDE API.

## **Comments**

TBD.

## **See Also**

[ICommenceConversation](#)

# **ICommenceConversation::Request**

**String Request**(String *pDDECommand*)

Processes the DDE Request.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CONVERSATION.D

## **Return Value**

Returns String value on success.

## **Parameters**

*pDDECommand*

Text with the DDE Request. Syntax is identical to the commands used by the DDE API.

## **Comments**

TBD.

## **See Also**

[ICommenceConversation](#)



# ICommenceCursor Object

Represents a cursor or query that is built to retrieve data

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## Properties

### String Category

(read-only) Name of the underlying Commence category.

### Long ColumnCount

(read-only) Number of columns in this cursor (-1 on error).

### Long RowCount

(read-only) Number of rows in this cursor (-1 on error).

### Boolean Shared

(read-only) TRUE if category is shared in a workgroup.

## Methods

### Boolean SetColumn(Long nColumn, String pName, Long nFlags)

Defines the column set for the cursor.

### Boolean SetFilter(String pFilter, Long nFlags)

Defines a filter clause for the cursor.

### Boolean SetLogic(String pLogic, Long nFlags)

Defines the filter logic for the cursor.

### Boolean SetSort(String pLogic, Long nFlags)

Defines the sort criteria for the cursor.

### Long SeekRow(Long bkOrigin, Long nRows)

Seek to a particular row in the cursor.

### Long SeekRowApprox(Long nNumerator, Long nDenom)

Seek to an approximate position in the cursor.

### ICommenceQueryRowSet GetQueryRowSet(Long nCount, Long nFlags)

Create a rowset object with the results of a query.

### ICommenceQueryRowSet GetQueryRowSetByID(String pRowID, Long nFlags)

Create a rowset object with a particular row loaded.

### ICommenceAddRowSet GetAddRowSet(Long nCount, Long nFlags)

Create a rowset of new items to add to the database.

### ICommenceEditRowSet GetEditRowSet(Long nCount, Long nFlags)

Create a rowset of existing items for editing.

### ICommenceEditRowSet GetEditRowSetByID(String pRowID, Long nFlags)

Create a rowset for editing a particular row.

### ICommenceDeleteRowSet GetDeleteRowSet(Long nCount, Long nFlags)

Create a rowset of existing items for deletion.

### ICommenceDeleteRowSet GetDeleteRowSetByID(String pRowID, Long nFlags)

Create a rowset for deleting a particular row.

### Boolean SetActiveItem

Set active item used for view cursors using a view linking filter.

### Boolean SetActiveDate

Set active date used for view cursors using a view linking filter.

### Boolean SetActiveDateRange

Set active date range used for view cursors using a view linking filter.

**Boolean SetRelatedColumn**

Adds a related (indirect/connected field) column to the cursor.

# ICommenceCursor::Category

**Data Type**  
**String**

**Description**

(read-only) Name of the underlying Commence category.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

**See Also**

[ICommenceCursor](#)

# **ICommenceCursor::ColumnCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of columns in this cursor (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **See Also**

[ICommenceCursor](#)

# **ICommenceCursor::RowCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of rows in this cursor (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **See Also**

[ICommenceCursor](#)

# ICommenceCursor::Shared

## Data Type

Boolean

## Description

(read-only) TRUE if category is shared in a workgroup.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## See Also

[ICommenceCursor](#)

# **ICommenceCursor::SetFilter**

**Boolean SetFilter(String *pFilter*, Long *nFlags*)**

Defines a filter clause for the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*pFilter*

Text defining the new filter clause. Syntax is identical to the one used by the DDE ViewFilter request (qv).

*nFlags*

Unused at present, must be 0.

## **Comments**

The *pFilter* string defines which filter clause is to be replaced. If the clause is already defined it will be overwritten.

If the cursor is opened in `CURSOR_VIEW` mode, the `SetFilter` only affects the cursor's secondary filter. That is, when building the rowset, the view's filter is first evaluated. Items that match are then passed through the cursor's secondary filter. The rowset only contains items that satisfy both filters.

## **See Also**

[ICommenceCursor](#)

[SetLogic](#)

# **ICommenceCursor::SetLogic**

**Boolean SetLogic(String *pLogic*, Long *nFlags*)**

Defines the filter logic for the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*pLogic*

Text defining the new filter logic. Syntax is identical to the one used by the DDE ViewConjunction request (qv).

*nFlags*

Unused at present, must be 0.

## **Comments**

Unless otherwise specified, the default logic is AND, AND, AND.

## **See Also**

[ICommenceCursor](#)

[SetFilter](#)



# **ICommenceCursor::SetSort**

**Boolean SetSort(String *pSort*, Long *nFlags*)**

Defines the sort criteria for the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*pSort*

Text defining the new sort criteria. Syntax is identical to the one used by the DDE ViewSort request (qv).

*nFlags*

Unused at present, must be 0.

## **Comments**

If the cursor is opened in CMC\_CURSOR\_VIEW mode, the sort defaults to the view's sort. All other cursor modes default to ascending sort by the Name field.

## **See Also**

[ICommenceCursor](#)

# **ICommenceCursor::SetColumn**

**Boolean SetColumn(Long *nColumn*, String *pName*, Long *nFlags*)**

Defines the column set for the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*nColumn*

The (0-based) index of the column to set.

*pName*

Name of the field to use in this column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_ALL - create column set of all fields

## **Comments**

When defining a column set, the columns must be defined in sequential order (0, 1, 2, etc.). This is to prevent problems with undefined columns (e.g. 0, 1, 3, ...).

Duplicate columns are not supported. Each column must map to a different field.

Not all Commence field types can be included in the cursor definition. The set of supported field types exactly matches those fields that can be displayed in a Commence report (minus combined fields and indirect fields).

## **See Also**

[ICommenceCursor](#)

[ICommenceDB::GetCursor](#)

# **ICommenceCursor::SeekRow**

**Long SeekRow(Long *bkOrigin*, Long *nRows*)**

Seek to a particular row in the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Actual number of rows moved, -1 on error.

## **Parameters**

*bkOrigin*

Position to move from.

Can be one of the following:

BOOKMARK\_BEGINNING (0) - from first row

BOOKMARK\_CURRENT (1) - from current row

BOOKMARK\_END (2) - from last row

*nRows*

Number of rows to move the current row pointer.

## **Comments**

For any cursor, there is a 'current row pointer'. When the cursor is created, this defaults to the first row.

SeekRow will reposition the current row pointer. GetQueryRowSet, GetEditRowSet and

GetDeleteRowSet will also advance the current row pointer.

## **See Also**

[ICommenceCursor](#)

[RowCount](#)

[SeekRowApprox](#)

[GetQueryRowSet](#)

[GetEditRowSet](#)

[GetDeleteRowSet](#)

# **ICommenceCursor::SeekRowApprox**

**Long SeekRowApprox(Long *nNumerator*, Long *nDenominator*)**

Seek to an approximate position in the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Actual number of rows moved, -1 on error.

## **Parameters**

*nNumerator*

Numerator for fractional position in the cursor.

*nDenominator*

Denominator for the fractional position in the cursor.

## **See Also**

[ICommenceCursor](#)

[RowCount](#)

[SeekRow](#)

[GetQueryRowSet](#)

[GetEditRowSet](#)

[GetDeleteRowSet](#)

# ICommenceCursor::GetQueryRowSet

**ICommenceQueryRowSet** GetQueryRowSet(Long *nCount*, Long *nFlags*)

Create a rowset object with the results of a query.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## Return Value

Returns Pointer to rowset object on success, NULL on error.

## Parameters

*nCount*

Maximum number of rows to retrieve.

*nFlags*

Unused at present, must be 0.

## Comments

The rowset inherits to column set from the cursor.

The cursor's 'current row pointer' determines the first row to be included in the rowset.

The returned rowset can have fewer than *nCount* rows (e.g. if the current row pointer is near the end).

Use ICommenceXRowSet::RowCount to determine the actual row count.

GetQueryRowSet will advance the 'current row pointer' by the number of rows in the rowset.

## See Also

[ICommenceCursor](#)

[ICommenceQueryRowSet](#)

[RowCount](#)

[SeekRow](#)

[SeekRowApprox](#)

# **ICommenceCursor::GetQueryRowSetByID**

**ICommenceQueryRowSet** **GetQueryRowSetByID**(String *pRowID*, Long *nFlags*)

Create a rowset object with a particular row loaded.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

## **Parameters**

*pRowID*

Unique ID string obtained from GetRowID().

*nFlags*

Unused at present, must be 0.

## **Comments**

The rowset inherits to column set from the cursor.

The cursor's 'current row pointer' is not advanced.

## **See Also**

[ICommenceCursor](#)

[ICommenceQueryRowSet::GetRowID](#)

# **ICommenceCursor::GetAddRowSet**

**ICommenceAddRowSet** **GetAddRowSet**(Long *nCount*, Long *nFlags*)

Create a rowset of new items to add to the database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

(Long *nCount*, Long *nFlags*)

## **Parameters**

*nCount*

Number of rows to create.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_SHARED - all rows default to shared

## **Comments**

The rowset inherits the column set from the cursor.

When first created, each row is initialized to field default values.

## **See Also**

[ICommenceCursor](#)

[ICommenceAddRowSet](#)

# **ICommenceCursor::GetEditRowSet**

**ICommenceEditRowSet GetEditRowSet(Long *nCount*, Long *nFlags*)**

Create a rowset of existing items for editing.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

## **Parameters**

*nCount*

Number of rows to retrieve.

*nFlags*

Unused at present, must be 0.

## **Comments**

The rowset inherits the column set from the cursor.

## **See Also**

[ICommenceCursor](#)

[ICommenceEditRowSet](#)



# **ISCommenceCursor::GetEditRowSetByID**

**ISCommenceEditRowSet** GetEditRowSetByID(String *pRowID*, Long *nFlags*)

Create a rowset for editing a particular row.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

## **Parameters**

*pRowID*

Unique ID string obtained from GetRowID().

*nFlags*

Unused at present, must be 0.

## **Comments**

The rowset inherits the column set from the cursor.

The cursor's 'current row pointer' is not advanced.

## **See Also**

[ISCommenceCursor](#)

[ISCommenceQueryRowSet::GetRowID](#)

[ISCommenceEditRowSet::GetRowID](#)

[ISCommenceDeleteRowSet::GetRowID](#)

# **ICommenceCursor::GetDeleteRowSet**

**ICommenceDeleteRowSet** **GetDeleteRowSet**(Long *nCount*, Long *nFlags*)

Create a rowset of existing items for deletion.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

## **Parameters**

*nCount*

Number of rows to retrieve.

*nFlags*

Unused at present, must be 0.

## **Comments**

The rowset inherits the column set from the cursor.

## **See Also**

[ICommenceCursor](#)

[ICommenceDeleteRowSet](#)

# **ICommenceCursor::GetDeleteRowSetByID**

**ICommenceDeleteRowSet** GetDeleteRowSetByID(String *pRowID*, Long *nFlags*)

Create a rowset for deleting a particular row.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns Pointer to rowset object on success, NULL on error.

## **Parameters**

*pRowID*

Unique ID string obtained from GetRowID().

*nFlags*

Unused at present, must be 0.

## **Comments**

The rowset inherits the column set from the cursor.

The cursor's 'current row pointer' is not advanced.

## **See Also**

[ICommenceCursor](#)

[ICommenceDeleteRowSet](#)

[ICommenceQueryRowSet::GetRowID](#)

[ICommenceEditRowSet::GetRowID](#)

[ICommenceDeleteRowSet::GetRowID](#)

# **ICommenceCursor::SetActiveItem**

**Boolean SetActiveItem(String *pCategoryName*, String *pRowID*, Long *flags*)**

Set active item used for view cursors using a view linking filter.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, else FALSE on error

## **Parameters**

*pCategoryName*

Category name of the active item used with view linking filter.

*pRowID*

Unique ID string obtained from GetRowID() indicating the active item used with view linking filter.

*flags*

Unused at present, must be 0.

## **Comments**

This method is used with a view cursor using view linking by active item. This method enables the active item to be set via the API, separate from the active item in the Commence UI.

## **See Also**

[ICommenceCursor](#)

[SetActiveDate](#)

[SetActiveDateRange](#)

[ICommenceQueryRowSet::GetRowID](#)

# **ICommenceCursor::SetActiveDate**

**Boolean SetActiveDate(String *sDate*, Long *flags*)**

Set active date used for view cursors using a view linking filter.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, else FALSE on error

## **Parameters**

*sDate*

Date value used with view linking filter; supports AI date values such as 'today'.

*flags*

Unused at present, must be 0.

## **Comments**

This method is used with a view cursor using view linking by active date. This method enables the active date to be set via the API, separate from the active date in the Commence UI.

## **See Also**

[ICommenceCursor](#)

[SetActiveItem](#)

[SetActiveDateRange](#)

# ICommenceCursor::SetActiveDateRange

**Boolean SetActiveDateRange(String startDate, String endDate, Long flags)**

Set active date range used for view cursors using a view linking filter.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## Return Value

Returns TRUE on success, else FALSE on error

## Parameters

*startDate*

Date value of start date used with view linking filter; supports AI date values such as 'today'.

*endDate*

Date value of end date used with view linking filter; supports AI date values such as 'next monday'.

*flags*

Unused at present, must be 0.

## Comments

This method is used with a view cursor using view linking by active date range. This method enables the active date range to be set via the API, separate from the active date range in the Commence UI.

## See Also

ICommenceCursor

SetActiveItem

SetActiveDate

# **ICommenceCursor::SetRelatedColumn**

**Boolean SetRelatedColumn(Long *nColumn*, String *pConnName*, String *pCatName*, String *pName*, Long *nFlags*)**

Adds a related (indirect/connected field) column to the cursor.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/CURSOR.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*nColumn*

The (0-based) index of the column to set.

*pConnName*

Name of the connection to use in this column.

*pCatName*

Name of the connected Category to use in this column.

*pName*

Name of the field in the connected category to use in this column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_ALL - create column set of all fields

## **Comments**

When defining a column set, the columns must be defined in sequential order (0, 1, 2, etc.). This is to prevent problems with undefined columns (e.g. 0, 1, 3, ...).

Duplicate columns are not supported. Each column must map to a different field.

Not all Commence field types can be included in the cursor definition. The set of supported field types exactly matches those fields that can be displayed in a Commence report (minus combined fields and indirect fields).

Sample call: SetRelatedColumn("Relates To", "History", "Date", 0)

This call will add the Date field to the cursor via the 'Relates to History' connection.

## **See Also**

[ICommenceCursor](#)

[ICommenceDB::GetCursor](#)

# ICommenceDB Object

Represents a Commence Database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## Properties

### String Name

(read-only) Name of the Commence database.

### String Path

(read-only) Full path of the Commence database.

### String RegisteredUser

(read-only) CR/LF delimited string with user name, company name and serial number.

### Boolean Shared

(read-only) TRUE if database is enrolled in a workgroup.

### String Version

(read-only) Version number in x.y format.

### String VersionExt

(read-only) Version number in x.y.z.w format.

## Methods

### ICommenceConversation GetConversation

Create a conversation object.

### ICommenceCursor GetCursor

Create a cursor object.



# **ICommenceDB::Name**

**Data Type**  
**String**

## **Description**

(read-only) Name of the Commence database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## **See Also**

[ICommenceDB](#)

# ICommenceDB::Path

**Data Type**

**String**

**Description**

(read-only) Full path of the Commence database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

**See Also**

[ICommenceDB](#)

# **ICommenceDB::RegisteredUser**

## **Data Type**

**String**

## **Description**

(read-only) CR/LF delimited string with user name, company name and serial number.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## **See Also**

[ICommenceDB](#)

# ICommenceDB::Shared

## Data Type

Boolean

## Description

(read-only) TRUE if database is enrolled in a workgroup.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## See Also

[ICommenceDB](#)

# ICommenceDB::Version

**Data Type**

**String**

**Description**

(read-only) Version number in x.y format.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

**See Also**

[ICommenceDB](#)

# **ICommenceDB::VersionExt**

**Data Type**

**String**

**Description**

(read-only) Version number in x.y.z.w format.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

**See Also**

[ICommenceDB](#)

# **ICommenceDB::GetConversation**

**ICommenceConversation GetConversation(String *pszApplicationName*, String *pszTopic*)**

Create a conversation object.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## **Return Value**

Returns a pointer to conversation object on success, NULL on error.

## **Parameters**

*pszApplicationName*

DDE Application name.

The only valid value is "Commence".

*pszTopic*

DDE Topic name, must be a valid Commence topic name.

The DDE.HLP file contains a reference to the allowed DDE topic names.

Examples include "GetData", "ViewData", etc.

## **See Also**

Related Methods and Objects

[ICommenceDB](#)

[ICommenceConversation](#)

[ICommenceConversation::Execute](#)

[ICommenceConversation::Request](#)

# ICommenceDB::GetCursor

**ICommenceCursor GetCursor(Long nMode, String pName, Long nFlags)**

Create a cursor object.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DB.D

## Return Value

Returns a pointer to cursor object on success, NULL on error.

## Parameters

*nMode*

Type of Commence data to access with this cursor.

Valid values are:

- 0 - CMC\_CURSOR\_CATEGORY  
Use the Commence category specified by pName.
- 1 - CMC\_CURSOR\_VIEW  
Use the Commence view specified by pName.
- 2 - CMC\_CURSOR\_PILOTAB  
Use the Commence category and fields defined from  
Preferences - Other Apps - 3Com Pilot Address Book
- 3 - CMC\_CURSOR\_PILOTMEMO  
Use the Commence category and fields defined from  
Preferences - Other Apps - 3Com Pilot Memo Pad
- 5 - CMC\_CURSOR\_PILOTTODO  
Use the Commence category and fields defined from  
Preferences - Other Apps - 3Com Pilot To Do List
- 6 - CMC\_CURSOR\_PILOTAPPT  
Use the Commence category and fields defined from  
Preferences - Other Apps - 3Com Pilot Date Book

*pName*

Name of an object in the database.

Use determined by nMode:

- For CMC\_CURSOR\_CATEGORY, pName is the category name.
- For CMC\_CURSOR\_VIEW, pName is the view name.
- For CMC\_CURSOR\_PILOTAB, pName is unused.
- For CMC\_CURSOR\_PILOTMEMO, pName is unused.
- For CMC\_CURSOR\_PILOTTODO, pName is unused.
- For CMC\_CURSOR\_PILOTAPPT, pName is unused.

*nFlags*

Addition option flags.

Logical OR of following option flags:

- CMC\_FLAG\_PILOT - Save Item agents defined for the Pilot  
subsystem will fire.
- CMC\_FLAG\_INTERNET - Save Itgem agents defined for the  
Internet/intranet will fire.

## Comments

For CMC\_CURSOR\_CATEGORY, the resulting cursor will have a column set composed of all supported fields in the category (in no particular order).

For CMC\_CURSOR\_VIEW, the resulting cursor will inherit the view's filter, sort and column set.



ICommenceCursor methods can be used to change these attributes.

For CMC\_CURSOR\_PILOT\*, the column set for the resulting cursor will only include fields defined by the Commence preferences (in no particular order). It is not possible to change the filter, sort or column set.

See the Developer Notes for more information about the CMC\_FLAG\_PILOT and CMC\_FLAG\_INTERNET flags.

**See Also**

Related Methods and Objects

[ICommenceDB](#)

[ICommenceCursor](#)

[ICommenceCursor::SetFilter](#)

[ICommenceCursor::SetLogic](#)

[ICommenceCursor::SetSort](#)

[ICommenceCursor::SetColumn](#)

# **ICommenceDeleteRowSet Object**

Represents the set of items to delete from the database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Properties**

### **Long ColumnCount**

(read-only) Number of columns in this rowset (-1 on error).

### **Long RowCount**

(read-only) Number of rows in this rowset (-1 on error).

## **Methods**

### **String GetRowValue**

Returns the field value at the given (row,column) in text form.

### **String GetColumnLabel**

Return the label associated with a particular column.

### **Long GetColumnIndex**

Search the column set and return the index of the column with the given label.

### **Long DeleteRow**

Mark a row for deletion.

### **Long Commit**

Make row modifications permanent (commit to disk).

### **String GetRow**

Returns an entire row's field values.

### **String GetRowID**

Returns a unique identifier for a row.

### **Boolean GetShared**

Return the row's current shared vs. local status.

# **ICommenceDeleteRowSet::ColumnCount**

## **Data Type**

Long

## **Description**

(read-only) Number of columns in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **See Also**

[ICommenceDeleteRowSet](#)

# **ICommenceDeleteRowSet::RowCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of rows in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **See Also**

[ICommenceDeleteRowSet](#)

# **ICommenceDeleteRowSet::GetRowValue**

**String GetRowValue(Long *nRow*, Long *nCol*, Long *nFlags*)**

Returns the field value at the given (row,column) in text form.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns Field value in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*nFlags*

option flags.

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

See the Developer Notes for more information about the canonical format.

## **See Also**

[ICommenceDeleteRowSet](#)

[RowCount](#)

[ColumnCount](#)

[GetRow](#)

# **ICommenceDeleteRowSet::GetColumnLabel**

**String GetColumnLabel(Long *nCol*, Long *nFlags*)**

Return the label associated with a particular column.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns Column label in text form on success, NULL on error.

## **Parameters**

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will return the labels used with the view.

Specify the CMC\_FLAG\_FIELD\_NAME to force the underlying Commence field name to be returned.

## **See Also**

ICommenceDeleteRowSet

GetColumnIndex

# **ICommenceDeleteRowSet::GetColumnIndex**

**Long GetColumnIndex(String *pLabel*, Long *nFlags*)**

Search the column set and return the index of the column with the given label.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns 0-based column index on success, -1 on error.

## **Parameters**

*pLabel*

The column label to map.

*nFlags*

option flags.

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will first search the view labels for a possible match. If not found, the field labels will then be searched. Specify the CMC\_FLAG\_FIELD\_NAME to force only the underlying Commence field name to be searched.

## **See Also**

[ICommenceDeleteRowSet](#)

[GetColumnLabel](#)

# **ICommenceDeleteRowSet::DeleteRow**

**Long DeleteRow(Long *nRow*, Long *nFlags*)**

Mark a row for deletion.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nFlags*

Unused at present, must be 0.

## **Comments**

Deletion is not permanent until Commit() is called.

## **See Also**

ICommenceDeleteRowSet

Commit



# **ICommenceDeleteRowSet::Commit**

**Long Commit**(Long *nFlags*)

Make row modifications permanent (commit to disk).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nFlags*

Unused at present, must be 0.

## **Comments**

After Commit(), the ICommenceDeleteRowSet is no Longer valid and should be discarded.

## **See Also**

ICommenceDeleteRowSet

DeleteRow

# **ICommenceDeleteRowSet::GetRow**

**String GetRow(Long *nRow*, String *pDelim*, Long *nFlags*)**

Returns an entire row's field values.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns Row values in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*pDelim*

Delimiter to use between field values.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

pDelim is used to separate field values. pDelim can be up to 20 chars.

Returned string is EOS terminated. Format is: <col1><delim><col2><delim>...<coln><EOS>

See Developer Notes for more information on canonical format.

## **See Also**

ICommenceDeleteRowSet

GetRowValue

# **ICommenceDeleteRowSet::GetRowID**

**String GetRowID(Long *nRow*, Long *nFlags*)**

Returns a unique identifier for a row.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns a unique ID string (less than 100 chars) on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nFlags*

Unused at present, must be 0.

## **Comments**

Unlike the row number, this ID value is valid across cursor sessions.

The 'scope' of the ID value is the Commence database from which it was retrieved. The ID value is not valid in other Commence databases, even if they are in the same workgroup and sync.

Do not make any assumptions about the contents of the returned ID string. Format may change in the future.

## **See Also**

[ICommenceDeleteRowSet](#)

[ICommenceCursor::GetEditRowSetByID](#)

[ICommenceCursor::GetDeleteRowSetByID](#)

# **ICommenceDeleteRowSet::GetShared**

**Boolean** **GetShared**(Long *nRow*)

Return the row's current shared vs. local status.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/DELETE.D

## **Return Value**

Returns TRUE if row is shared, FALSE otherwise.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **See Also**

[ICommenceDeleteRowSet](#)

# **ICommenceEditRowSet Object**

Represents the set of items to edit in the database.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Properties**

### **Long ColumnCount**

(read-only) Number of columns in this rowset (-1 on error).

### **Long RowCount**

(read-only) Number of rows in this rowset (-1 on error).

## **Methods**

### **String GetRowValue**

Returns the field value at the given (row,column) in text form.

### **String GetColumnLabel**

Return the label associated with a particular column.

### **Long GetColumnIndex**

Search the column set and return the index of the column with the given label.

### **Long ModifyRow**

Modify a field value in the rowset.

### **Long Commit**

Make row modifications permanent (commit to disk).

### ICommenceCursor **CommitGetCursor**

Make row modifications permanent (commit to disk) and return a cursor for the newly added data.

### **String GetRow**

Returns an entire row's field values.

### **Boolean GetShared**

Return the row's current shared vs. local status.

### **Boolean SetShared**

Mark a row to be shared.

### **String GetRowID**

Returns a unique identifier for a row.

# **ICommenceEditRowSet::ColumnCount**

## **Data Type**

Long

## **Description**

(read-only) Number of columns in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **See Also**

[ICommenceEditRowSet](#)

# **ICommenceEditRowSet::RowCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of rows in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **See Also**

[ICommenceEditRowSet](#)

# **ICommenceEditRowSet::GetRowValue**

**String GetRowValue(Long *nRow*, Long *nCol*, Long *nFlags*)**

Returns the field value at the given (row,column) in text form.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns Field value in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

See the Developer Notes for more information about the canonical format.

## **See Also**

[ICommenceEditRowSet](#)

[RowCount](#)

[ColumnCount](#)

[GetRow](#)



# **ICommenceEditRowSet::GetColumnLabel**

**String GetColumnLabel(Long *nCol*, Long *nFlags*)**

Return the label associated with a particular column.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns Column label in text form on success, NULL on error.

## **Parameters**

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will return the labels used with the view.

Specify the CMC\_FLAG\_FIELD\_NAME to force the underlying Commence field name to be returned.

## **See Also**

[ICommenceEditRowSet](#)

[GetColumnIndex](#)

# **ICommenceEditRowSet::GetColumnIndex**

**Long GetColumnIndex(String *pLabel*, Long *nFlags*)**

Search the column set and return the index of the column with the given label.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns 0-based column index on success, -1 on error.

## **Parameters**

*pLabel*

The column label to map.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will first search the view labels for a possible match. If not found, the field labels will then be searched. Specify the CMC\_FLAG\_FIELD\_NAME to force only the underlying Commence field name to be searched.

## **See Also**

[ICommenceEditRowSet](#)

[GetColumnLabel](#)

# **ICommenceEditRowSet::ModifyRow**

**Long ModifyRow(Long *nRow*, Long *nCol*, String *pBuf*, Long *nFlags*)**

Modify a field value in the rowset.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*pBuf*

New field value in text form.

*nFlags*

Unused at present, must be 0.

## **Comments**

Modifications to the rowset will be reflected by GetRowValue() and GetRow() but changes are not permanent until Commit() is called.

## **See Also**

ICommenceEditRowSet

Commit

CommitGetCursor

# **ICommenceEditRowSet::Commit**

## **Long Commit(Long *nFlags*)**

Make row modifications permanent (commit to disk).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns 0 on success, -1 on error.

## **Parameters**

*nFlags*

Unused at present, must be 0.

## **Comments**

After Commit(), the ICommenceEditRowSet is no Longer valid and should be discarded.

## **See Also**

[ICommenceEditRowSet](#)

[ModifyRow](#)

[CommitGetCursor](#)

# **ICommenceEditRowSet::CommitGetCursor**

**ICommenceCursor CommitGetCursor(Long *nFlags*)**

Make row modifications permanent (commit to disk) and return a cursor for the newly added data.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns ICommenceCursor object on success, NULL on error.

## **Parameters**

*nFlags*

Unused at present, must be 0.

## **Comments**

After Commit(), the ICommenceEditRowSet is no Longer valid and should be discarded.

## **See Also**

[ICommenceEditRowSet](#)

[ModifyRow](#)

[Commit](#)

# **ICommenceEditRowSet::GetRow**

**String GetRow(Long *nRow*, String *pDelim*, Long *nFlags*)**

Returns an entire row's field values.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns Row values in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*pDelim*

Delimiter to use between field values.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

pDelim is used to separate field values. pDelim can be up to 20 chars.

Returned string is EOS terminated. Format is: <col1><delim><col2><delim>...<coln><EOS>

See the Developer Notes for more information about the canonical format.

## **See Also**

ICommenceEditRowSet

GetRowValue

# **ICommenceEditRowSet::GetShared**

**Boolean GetShared(Long *nRow*)**

Return the row's current shared vs. local status.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns TRUE if row is shared, FALSE otherwise.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **See Also**

[ICommenceEditRowSet](#)  
[SetShared](#)

# **ICommenceEditRowSet::SetShared**

**Boolean SetShared(Long *nRow*)**

Mark a row to be shared.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns TRUE on success, FALSE on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **See Also**

ICommenceEditRowSet

GetShared



# **ICommenceEditRowSet::GetRowID**

**String** GetRowID(Long *nRow*, Long *nFlags*)

Returns a unique identifier for a row.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/EDIT.D

## **Return Value**

Returns a unique ID string (less than 100 chars) on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nFlags*

Unused at present, must be 0.

## **Comments**

Unlike the row number, this ID value is valid across cursor sessions.

The 'scope' of the ID value is the Commence database from which it was retrieved. The ID value is not valid in other Commence databases, even if they are in the same workgroup and sync.

Do not make any assumptions about the contents of the returned ID string. Format may change in the future.

## **See Also**

[ICommenceEditRowSet](#)

[ICommenceCursor::GetQueryRowSetByID](#)

[ICommenceCursor::GetEditRowSetByID](#)

[ICommenceCursor::GetDeleteRowSetByID](#)

## Developer Notes

The following topics contain technical notes about the Database API.

3Com Pilot support

Canonical Format

CMC\_FLAG\_PILOT

CMC\_FLAG\_INTERNET

# Constants

## Flags used with GetCursor

CMC\_CURSOR\_CATEGORY = 0  
open based on a category

CMC\_CURSOR\_VIEW = 1  
open based on a view

CMC\_CURSOR\_PILOTAB = 2  
3Com Pilot Address Book

CMC\_CURSOR\_PILOTMEMO= 3  
3Com Pilot Memo

CMC\_CURSOR\_PILOTTODO= 5  
3Com Pilot to-do list

CMC\_CURSOR\_PILOTAPPT= 6  
3Com Pilot appt calendar

CMC\_CURSOR\_OUTLOOKAB= 7  
MS Outlook contacts preference

CMC\_CURSOR\_OUTLOOKAPPT=8  
MS Outlook calendar preference

CMC\_CURSOR\_EMAILLOG=9  
MS Outlook Email Log preference

CMC\_CURSOR\_OUTLOOKTASK=10  
MS Outlook Task preference

CMC\_CURSOR\_MERGE=11  
open based on the view data used with the Send Letter command

## Flags used with SeekRow

BOOKMARK\_BEGINNING = 0  
Offset from beginning of the rowset.

BOOKMARK\_CURRENT = 1  
Offset from current position.

BOOKMARK\_END = 2  
Offset from end of the rowset.

## Option Flags

CMC\_FLAG\_FIELD\_NAME = &H0001  
return actual field name

CMC\_FLAG\_ALL = &H0002  
include all fields

CMC\_FLAG\_SHARED = &H0004  
init as shared

CMC\_FLAG\_PILOT = &H0008  
changes from 3Com Pilot

CMC\_FLAG\_CANONICAL = &H0010  
data in canonical format

CMC\_FLAG\_INTERNET = &H0020  
changes from Internet

## 3Com Pilot support

For CMC\_CURSOR\_PILOTx cursor modes, Commence will use the 'Sync Condition' field (which must be a checkbox or a connection to the "(-me-)" item) to determine which items appear in the cursor. The column set only includes fields for which mappings exist.

For example, if the user did not select a Fax Number field, then it will not appear in the column set. This means `GetColumnIndex("Fax Number", 0)` can and will fail.

If 'Sync Condition' is not mapped to a Commence field, then all items in the category will appear in the cursor.

# Canonical Format

By default, data values are returned formatted according to the local Control Panel settings. Where applicable, specifying CMC\_FLAG\_CANONICAL will return data in a consistent format.

Data Type	Format	Notes
-----------	--------	-------

Date	yyyymmdd	
------	----------	--

Time	hh:mm	military time, 24 hour clock
------	-------	------------------------------

Number	123456.78	no separator, period for decimal delimiter
--------	-----------	--

Checkbox	TRUE or FALSE	English
----------	---------------	---------

## **CMC\_FLAG\_PILOT**

The Save Item trigger can be configured to fire only if an item is added or edited by specific subsystems. As part of Pilot support, the Save Item Options dialog box will include a Pilot Link check box (SS\_PILOT).

By default, a cursor in CMC\_CURSOR\_PILOTx mode will assume SS\_PILOT mode. When using other cursor modes, specifying CMC\_FLAG\_PILOT will force SS\_PILOT mode.

## **CMC\_FLAG\_INTERNET**

The Save Item trigger can be configured to fire only if an item is added or edited by specific subsystems. As part of Commence Web support, the Save Item Options dialog box will include a Internet/Intranet check box (SS\_INTERNET).

Specifying CMC\_FLAG\_INTERNET will allow SS\_INTERNET agents to fire.



# ICommenceQueryRowSet Object

Represents a result set from a query

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## Properties

### Long ColumnCount

(read-only) Number of columns in this rowset (-1 on error).

### Long RowCount

(read-only) Number of rows in this rowset (-1 on error).

## Methods

### String GetRowValue

Returns the field value at the given (row,column) in text form.

### String GetColumnLabel

Return the label associated with a particular column.

### Long GetColumnIndex

Search the column set and return the index of the column with the given label.

### String GetRow

Returns an entire row's field values.

### String GetRowID

Returns a unique identifier for a row.

### Boolean GetShared

Return the row's current shared vs. local status.

### Long GetFieldToFile

Save the field value at the given (row,column) to a file.

# **ICommenceQueryRowSet::ColumnCount**

## **Data Type**

Long

## **Description**

(read-only) Number of columns in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **See Also**

[ICommenceQueryRowSet](#)

# **ICommenceQueryRowSet::RowCount**

## **Data Type**

**Long**

## **Description**

(read-only) Number of rows in this rowset (-1 on error).

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **See Also**

[ICommenceQueryRowSet](#)

# **ICommenceQueryRowSet::GetRowValue**

**String GetRowValue(Long *nRow*, Long *nCol*, Long *nFlags*)**

Returns the field value at the given (row,column) in text form.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns Field value in text form on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

See the Developer Notes about canonical format.

## **See Also**

[ICommenceQueryRowSet](#)

[RowCount](#)

[ColumnCount](#)

[GetRow](#)

[GetFieldToFile](#)

# **ICommenceQueryRowSet::GetColumnLabel**

**String GetColumnLabel(Long *nCol*, Long *nFlags*)**

Return the label associated with a particular column.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns Column label in text form on success, NULL on error.

## **Parameters**

*nCol*

The (0-based) index of the column.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will return the labels used with the view.

Specify the CMC\_FLAG\_FIELD\_NAME to force the underlying Commence field name to be returned.

## **See Also**

[ICommenceQueryRowSet](#)

[GetColumnIndex](#)

# **ICommenceQueryRowSet::GetColumnIndex**

**Long GetColumnIndex(String *pLabel*, Long *nFlags*)**

Search the column set and return the index of the column with the given label.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns 0-based column index on success, -1 on error.

## **Parameters**

*pLabel*

The column label to map.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_FIELD\_NAME - return field label (ignore view labels)

## **Comments**

If the cursor is created with CMC\_CURSOR\_VIEW, this will first search the view labels for a possible match. If not found, the field labels will then be searched. Specify the CMC\_FLAG\_FIELD\_NAME to force only the underlying Commence field name to be searched.

## **See Also**

[ICommenceQueryRowSet](#)

[GetColumnLabel](#)

# **ICommenceQueryRowSet::GetRow**

**String GetRow(Long *nRow*, String *pDelim*, Long *nFlags*)**

Returns an entire row's field values.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns Row values in text form on success, NULL on error.

(Long *nRow*, String *pDelim*, Long *nFlags*)

## **Parameters**

*nRow*

The (0-based) index of the row.

*pDelim*

Delimiter to use between field values.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## **Comments**

*pDelim* is used to separate field values. *pDelim* can be up to 20 chars.

Returned string is EOS terminated. Format is: <col1><delim><col2><delim>...<coln><EOS>

See Developer Notes for more information about the canonical format.

## **See Also**

[ICommenceQueryRowSet](#)

[GetRowValue](#)

[GetFieldToFile](#)

# **ICommenceQueryRowSet::GetRowID**

**String GetRowID(Long *nRow*, Long *nFlags*)**

Returns a unique identifier for a row.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns a unique ID string (less than 100 chars) on success, NULL on error.

## **Parameters**

*nRow*

The (0-based) index of the row.

*nFlags*

Unused at present, must be 0.

## **Comments**

Unlike the row number, this ID value is valid across cursor sessions.

The 'scope' of the ID value is the Commence database from which it was retrieved. The ID value is not valid in other Commence databases, even if they are in the same workgroup and sync.

Do not make any assumptions about the contents of the returned ID string. Format may change in the future.

Not yet supported but the unique ID will be useful in the future to edit/delete a particular item.

## **See Also**

[ICommenceQueryRowSet](#)



# **ICommenceQueryRowSet::GetShared**

**Boolean GetShared(Long *nRow*)**

Return the row's current shared vs. local status.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## **Return Value**

Returns TRUE if row is shared, FALSE otherwise.

## **Parameters**

*nRow*

The (0-based) index of the row.

## **See Also**

[ICommenceQueryRowSet](#)

# ICommenceQueryRowSet::GetFieldToFile

**Long GetFieldToFile(Long *nRow*, Long *nCol*, String *filename*, Long *nFlags*)**

Save the field value at the given (row,column) to a file.

Defined in: E:/DEV/PIM80/DOCS/DBAPI/PIMOA/QUERY.D

## Return Value

Returns the file size in bytes, or -1 on error.

## Parameters

*nRow*

The (0-based) index of the row.

*nCol*

The (0-based) index of the column.

*filename*

Filename and path where the field value is written.

*nFlags*

option flags

Logical OR of following option flags:

CMC\_FLAG\_CANONICAL - return field value in canonical form

## Comments

If there is no data in the indicated column, the return code is 0 and a file is not written. Otherwise, a file is created with the field's data and the size of the data is returned.

See the Developer Notes about canonical format.

## See Also

[ICommenceQueryRowSet](#)

[GetRowValue](#)

[GetRow](#)

