

## Basics: Use Macros

Last updated: May 2021



OCLC Online Computer Library  
Center, Inc.  
6565 Kilgour Place  
Dublin, OH 43017-3395

## Review History

Date	Section title	Description of changes
September 2011	1 Introduction	<ul style="list-style-type: none"><li>• New OCLC-supplied macro Add33x added to OCLC.mbk</li><li>• Legacy Help files (*.hlp) on Standard BASIC and Macro Editor and Debugger are unavailable in Connexion client under Windows Vista <b>and higher</b> versions.</li></ul>
	4 List of macro commands	New macro command, InsertFromCitedRecord, added to lists.
	7 Commands: Edit records	Detailed description and an example for InsertFromCitedRecord added.
March 2012	1 Introduction	Added descriptions of two new OCLC-supplied macros: <ul style="list-style-type: none"><li>• ClearELv3Workform</li><li>• GenerateERecord (this macro replaces two OCLC macros – GenerateESerial and GenerateESerialCONSER)</li></ul>
	4 List of macro commands	Added a new macro – GetClassNumber.
	7 Commands: Edit records	<ul style="list-style-type: none"><li>• Added 800, 810, 811, or 830 to the list of fields that can be used for the command InsertFromCitedRecord</li><li>• Added a detailed description of the new macro command, InsertFromCitedRecord</li></ul>
September 2012	All	Removed references to pathfinder.
April 2013	All	Removed references to Heritage Printed Book and Chinese Name Authority.
June 2016	All	Removed references to institution records.
May 2021	1 Introduction	Added note pointing to Connexion client version 3.0 documentation.
	7 Commands: Edit records	Updated description for ChangeRecordType.

© 2016 OCLC Online Computer Library Center, Inc.  
6565 Kilgour Place  
Dublin, OH 43017-3395 USA

The following OCLC product, service and business names are trademarks or service marks of OCLC, Inc.: CatExpress, Connexion, DDC, Dewey, Dewey Decimal Classification, OCLC, WorldCat, WorldCat Resource Sharing and "The world's libraries. Connected."

In addition, the WorldCat symbol and OCLC symbol are service marks of OCLC. Third-party product and service names are trademarks or service marks of their respective owners.

OCLC grants permission to photocopy this publication as needed.

## Contents

1. Introduction to OCLC Macro Language for Connexion client .....	8
Why use macros? .....	8
Client macro editor/debugger and general BASIC OCLC macro language Help .....	8
OCLC-supplied object class and variables .....	9
Why use an object?.....	9
How to use a client object.....	9
Syntax for client macro commands.....	9
Return values from client macro commands.....	10
OCLC-supplied macros.....	11
2. Use Connexion client macros .....	15
Client macro tools .....	15
How to create macros .....	16
Write macros.....	16
Tips for using the Macro Editor.....	17
View or edit macros.....	18
Check macro syntax.....	18
Macro example .....	19
Include macros in another macro .....	19
Run macros.....	20
Stop macros.....	20
Delete macros.....	21
Organize macros and macro books.....	21
Caution: OCLC-supplied macro books may be overwritten at upgrade .....	22
Actions not available for macros .....	22
3. Record macros (create automatically).....	24
Record a macro.....	24
Tips for using the macro recorder .....	25
4. List of Connexion client macro commands.....	26
List of client macro commands by function.....	26
Alphabetical list of macro commands .....	32
5. Connexion client macro commands: Log on and off .....	34
Log on.....	34
Log off.....	34
IsOnline.....	35
6. Connexion client macro commands: Search.....	36
Search .....	36

Search guidelines, indexes, examples .....	37
Browse.....	45
Browse Authority Expanded .....	46
7. Connexion client macro commands: Edit records .....	48
AddField.....	48
AddFieldLine .....	49
ControlHeadingsAll .....	50
UncontrolHeading .....	51
UncontrolAll.....	52
IsHeadingControlled.....	52
Copy .....	53
Cut .....	54
Paste.....	54
CopySelected.....	55
CutSelected .....	55
CopyField.....	56
CopyControlNumber .....	56
CursorPosition .....	57
DeleteCurrentField.....	57
DeleteCurrentWord .....	58
DeleteToEndOfCell .....	59
MoveFieldUp.....	59
MoveFieldDown .....	60
DeleteField.....	60
DeleteFieldLine .....	61
GetField .....	62
GetFieldLine.....	63
GetFieldUnicode .....	64
GetFieldLineUnicode.....	64
SetField.....	65
SetFieldLine .....	66
SetFixedField .....	67
GetFixedField.....	68
FixedFieldPosition.....	69
SetCursorFixedField .....	70
GetSelectedText .....	70
SetSelectedText.....	71
InsertText.....	71

FindText.....	72
ReplaceText.....	73
ReplaceTextAll.....	75
FindReplaceNextText.....	76
Cancel Changes .....	77
Reformat.....	78
Validate.....	78
CursorColumn.....	79
CursorRow.....	80
InsertMode.....	81
ApplyOnlineDefaultCD .....	82
ApplyOnlineCDBByName.....	82
ApplyCDFromList.....	83
ApplyLocalDefaultCD.....	84
ApplyLocalCDBByName.....	85
ChangeRecordType.....	86
InsertFromCitedRecord.....	87
GetClassNumber.....	87
8. Connexion client macro commands: Create records.....	89
DeriveNewRecord.....	89
OpenWorkform.....	90
OpenWorkFormForDatabase.....	91
9. Connexion client macro commands: Take actions on records.....	92
UpdateHoldings .....	92
ProduceAndUpdateHoldings.....	93
AltProduceAndUpdateHoldings.....	94
DeleteHoldings.....	95
Lock.....	96
ReleaseRecordLock.....	97
ReplaceRecord.....	98
ReplaceAndUpdateHoldings.....	99
SaveOnline.....	100
SaveToLocalFile.....	100
DeleteRecord.....	102
Export.....	102
ExportAppend.....	103
AddToAuthorityFile.....	104
AddToOnlineDatabase.....	105

SubmitForReview.....	105
RecallSubmitted.....	106
10. Connexion client macro commands: Work with records, lists, and files .....	108
SearchList.....	108
GetListItem.....	109
GetFirstSelectedItem .....	109
GetNextSelectedItem .....	110
GetFirstItem .....	110
GetNextItem.....	111
GetPrevItem.....	111
GetNext100Records.....	112
GetPrevious100Records .....	113
CloseList.....	114
CloseRecord .....	114
GetWindowTitle.....	115
ItemType.....	116
SetTopWindow.....	117
WindowCount.....	118
CreateLocalFile .....	119
CopyRecord .....	120
MoveRecord.....	120
Macro commands for navigating in records.....	121
11. Connexion client macro commands: Print and work with macros, statuses, holdings, labels, user profiles .....	122
SetMyStatus.....	122
SetWorkflowStatus.....	122
IsHeldBy .....	123
Holdings.....	124
Print .....	124
PrintList.....	125
PrintLabel.....	126
PrintLabelEx.....	127
GetLabelSpine .....	129
GetLabelSpineUnformatted.....	130
GetLabelCallNumber.....	130
GetLabelAuthor.....	131
GetLabelTitle.....	132
StartLabelBatch .....	132

EndLabelBatch.....	133
PrintMacro .....	133
RunMacro .....	133
AssignMacroToKey .....	134
QueryRecordStatus.....	134
LoadProfile.....	136
SendKeys .....	137
12. ASCII and hexadecimal codes for diacritics and special characters .....	140
Enter diacritics and special characters using macros .....	140

## 1. Introduction to OCLC Macro Language for Connexion client

**Note:** If you are using Connexion client version 3.0 or newer, see [Connexion client](#) for applicable documentation.

Macros are short programs you write to automate routine tasks. Macros are built from commands. Each command does one step in the procedure you want to run automatically. Creating and using macros is **optional** and requires some understanding of BASIC programming language. The client supports macros written using OCLC Macro Language (OML), which provides:

- Standard BASIC functions and statements
- Macro commands developed specifically for the client

### Why use macros?

Using macros can help to:

- Reduce the number of keystrokes and mouse clicks by running repetitive processing steps automatically
- Standardize procedures to reduce errors and inconsistencies
- Decrease staff time required for routine tasks

### Client macro editor/debugger and general BASIC OCLC macro language Help

In addition to information sources listed above, the client provides several online Help systems to assist with creating client macros.

To open Help topics for the macro editor/debugger and/or for the general BASIC OCLC macro language:

1. Click **Tools > Macros > Manage**. The Macro window opens.
2. Create a new macro.  
Or  
Select a macro in the displayed list and click **Edit**.

3. In the Macro Editor and Debugger window that opens, click  on the toolbar to open the Editor/Debugger Help window.

#### Caution for Windows Vista and Windows 7 users:

- These older Help files (\*.hlp) are unavailable under Windows Vista and Windows 7 unless you download the discontinued Help engine **WinHlp32.exe** from the Microsoft Web site at: <http://support.microsoft.com/kb/917607>.

See also known issues with WinHlp32.exe and workarounds at:  
<http://support.microsoft.com/kb/917607#knownissueupdate>

- In the Connexion client, only the Standard BASIC guide and Macro Editor and Debugger Help topics are affected. These topics are in two legacy Help files, **osbi5x.hlp** and **osbi5x.hlp**.

**Note:** These older .hlp files are located in the same folder as the main client Help file (clienthelp.chm):

**X:\Program Files\OCLC\Connexion\Program** (where **X** is the letter of your hard drive)

- Client macro command-specific Help in the main help file (**clienthelp.chm**) is not affected.

To get help with client-specific macro commands, click **Help > Client Help**, or click , or press <F1>. In the Help Contents tab, expand Basics/Use Macros.

**Reference table:** For diacritics and special characters in data used with macro commands such as *CS.GetField* or *CS.SetField*, see a list of codes, “ASCII and hexadecimal codes for diacritics and special characters.”

## OCLC-supplied object class and variables

- OCLC supplies the object class **Connex.Client**. Commands in OML macros automate Connexion client actions by accessing a subset of client functionality through this object class, which is the interface between the client functionality and OML.
- To gain access to the functions provided by **Connex.Client**, an OML macro contains two lines that do the following:
  - Define the object variable *CS* by which to reference **Connex.Client**: *Dim CS as Object*.
  - Set *CS* to reference an instance of the object class **Connex.Client**: *Set CS = CreateObject(“Connex.Client”)*.

## Why use an object?

The only way to create a macro that automates Connexion client operations is to use the variable *CS* as an instance of the class *Connex.Client*. You can then use the capabilities of that class of object to make the macro work. By using the appropriate object, you can, for example, write a macro to log on to Connexion, automate a search, modify a record, display a record from a list, or take an action on a record.

## How to use a client object

The tasks an object can do are called *methods* or *member functions* of the object. OML Help uses the term *commands* for the methods of the client object class.

For a list of commands for use in client macros, see “List of Connexion client macro commands.”

## Syntax for client macro commands

- Command syntax links client commands to the *Connex.Client* object.
- At the beginning of a macro, define the object variable *CS* to represent the object.
- For each macro command, add the prefix *CS.* to the command name to reference the required object through its predefined object variable, *CS*.

**Example:**

- *dim CS as Object*
  - *set CS = CreateObject(“Connex.Client”)*
  - *CS.WindowCount*  
(Returns the number of open record and list windows)
- At the beginning of a macro, name and declare data types for variables to represent the values you supply as arguments (for example, line numbers and text strings) for carrying out the commands.

**Example:**

- *dim CS as Object*
- *dim sData as String*
- *dim n as integer*
- *set CS = CreateObject("Connex.Client")*  
(Sets *sData* as a string of characters and *n* as an integer)

## Return values from client macro commands

- Many client macro commands are functions that return a value of TRUE (if the command executes successfully) or FALSE (if the command fails).
- For these commands to work correctly, you must assign a variable to store the return values. As each command executes, the stored value is updated.
- Optional. You can include *If . . Then . . Else* statements or other code that checks the value of this variable and, based on the result, displays messages or branches to different sections of the macro.  
*If...Then...Else* executes alternative blocks of program code based on one or more expressions.

### Syntax A

*If condition Then then\_statement [Else else\_statement] [bracketed parameter is optional]*

### Syntax B

*If condition Then*  
*[Else expression Then statement\_block]*  
*[Else statement\_block]*  
*End If*

Where:

- *condition* is any expression that evaluates to TRUE (non-zero) or FALSE (zero)
- *then\_statement* is any valid single expression
- *else\_statement* is any valid single expression
- *expression* is any expression that evaluates to TRUE (non-zero) or FALSE (zero)
- *statement\_block* is 0 or more valid expressions, separated by colons (:) or entered on separate lines

**Note:** When multiple statements are required in either the *Then* or *Else* clauses, use the block version (Syntax B) of the *If* statement.

- Use the following syntax to specify the variable **BOOL** to store the return value of each command:

**BOOL = CS.[command]**

**Note:** Standard OML commands to not use the *CS.* prefix.

### Examples:

**Set**  
**If . . Then . . Else**  
**MsgBox**

## OCLC-supplied macros

OCLC supplies several widely used macros:

- **FourFig** and **Sanborn** in a macro book file named **Dewey.mbk** for adding a Cutter number to a Dewey classification number. See instructions for using the Cutter macro in Cataloging, Print Labels and Accessions Lists, "Print labels."
- **GenerateAuthorityRecord** named **OCLC.mbk** for generating an authority record from a name heading in a bibliographic record. You can run the macros from **Tools > Macros > Manage**.

OCLC provides additional macros. The following table lists all OCLC-supplied macros supplied in **OCLC.mbk**, describes the purpose of each, and gives details for using them:

OCLC macro name and purpose	How to use
<p><b>Add&amp;Edit007</b></p> <p>Adds or edits field 007 in a bibliographic record via a menu</p>	<p>To add field 007:</p> <p>Retrieve a bibliographic record and run the macro, but do not place the cursor within any existing 007 field.</p> <p>To edit field 007:</p> <p>Retrieve a bibliographic record and run the macro after placing the cursor within any existing 007 field to be edited.</p>
<p><b>Add33x</b></p> <p>Adds 33x fields in a bibliographic or authority record via a menu</p>	<p>To add field 336, 337, and/or 338, retrieve a bibliographic or authority record and run the macro.</p>
<p><b>AddAuthority6xx</b></p> <p>Adds a 6xx field template to an authority record via a menu</p>	<p>To add a 6xx field template</p> <ol style="list-style-type: none"> <li>1. Retrieve an authority record or workform and run the macro.</li> <li>2. Choose an appropriate field from the menu and click <b>OK</b>.</li> </ol> <p><b>Result:</b> The macro adds the field to the record and places a fill character at the point in the field where it needs to be reviewed and further edited.</p>
<p><b>AddSerialNotes</b></p> <p>Adds a note field template to a serial bibliographic record via a menu</p>	<p>To add a note field template</p> <ol style="list-style-type: none"> <li>1. Retrieve an serial bibliographic record or workform and run the macro.</li> <li>2. Choose an appropriate field from the menu and click <b>OK</b>.</li> </ol> <p><b>Result:</b> The macro adds the field to the record and places a fill character at the point in the field where it needs to be reviewed and further edited.</p>

OCLC macro name and purpose	How to use
<p><b>BrowseAuthorityIndex</b></p> <p>Begins a browse search in the authority indexes based on a heading in a displayed authority or bibliographic record</p>	<p>To begin an authority browse search:</p> <p>Retrieve an authority or bibliographic record or workform, place the cursor in a heading or title field you want to browse, and run the macro.</p>
<p><b>BrowseBibliographicIndex</b></p> <p>Begins a browse search in the bibliographic indexes based on a heading in a displayed authority or bibliographic record</p>	<p>To begin a bibliographic browse search:</p> <p>Retrieve an authority or bibliographic record or workform, place the cursor in a heading or title field you want to browse, and run the macro.</p>
<p><b>ClearELvl3Workform</b></p> <p>Clears candidate fields from an encoding level 3 bibliographic record and replaces them with workform prompts. The prompts are a blend of two sets of workforms—AACR2 versus RDA.</p>	<p>To clear candidate fields, retrieve the encoding level 3 bibliographic record corresponding to an item to be cataloged and run the macro.</p>
<p><b>Generate034</b></p> <p>Generates field 034 based on data in field 255 in the same bibliographic record</p>	<p>To generate field 034:</p> <p>Retrieve a bibliographic record with field 255 and run the macro.</p> <p><b>Note:</b> Field 255 must be correctly formulated per AACR2 rules and must use the degree sign, the miagkii znak, and the tverdyi znak to indicate degrees, minutes, and seconds in any coordinates.</p>
<p><b>Generate043</b></p> <p>Generates field 043 based on data in 6xx fields in the same authority or bibliographic record</p>	<p>To generate field 043:</p> <p>Retrieve an authority record or workform that contains field 151 and/or 78, or retrieve a bibliographic record or workform that contains LCSH, LC Children's, MeSH, or Canadian subject headings, and run the macro.</p>
<p><b>Generate891</b></p> <p>Generates a pair of 891 fields based on frequency and numbering data in the same bibliographic record</p>	<p>To generate a pair of 891 fields:</p> <ol style="list-style-type: none"> <li>1. Retrieve a serial bibliographic record or completed workform and run the macro.</li> <li>2. Edit the resulting fields as appropriate.</li> </ol>

OCLC macro name and purpose	How to use
<p><b>GenerateAuthorityRecord</b></p> <p>Generates a name authority workform based on a heading in a displayed bibliographic record (see detailed instructions in Authorities, Create Authority Records, “Generate an authority record from a name heading in a bibliographic record...”)</p>	<p>To generate an authority record:</p> <p>Place the cursor within a valid heading field of a bibliographic record and run the macro.</p>
<p><b>GenerateERecord</b></p> <p>Creates a workform to represent an online version of an item based on the print, microform, or another format version of a bibliographic record for the same title</p>	<p>To create an online version workform:</p> <p>Retrieve a bibliographic record for the same title and run the macro.</p> <p><b>Note:</b> This macro replaces the following two former OCLC macros: GenerateESerial and GenerateESerialCONSER.</p>
<p><b>InsertHeading</b></p> <p>Inserts a displayed authority record heading into a previously displayed bibliographic record heading field</p>	<p>To insert a heading:</p> <ol style="list-style-type: none"> <li>1. Retrieve a bibliographic record and place the cursor in a heading field.</li> <li>2. Search the authority file manually or via the BrowseAuthorityIndex macro for the corresponding heading.</li> <li>3. Once the appropriate authority record is displayed, run the macro.</li> </ol> <p><b>Result:</b> The macro compares the displayed authority heading with the heading in the previously displayed bibliographic record and inserts it into the bibliographic record if there are differences</p>

OCLC may supply additional macros or modify existing macros in the **OCLC.mbk** book file in future versions of the client

**Caution for retaining your own macros when upgrading to the next client version:**

- When OCLC adds or modifies macros in Dewey.mbk and/or OCLC.mbk, these files are overwritten when you upgrade to a new version of the client.
- You will lose your customized macros if you store them in either of these files when you upgrade. Copy any of your own macros to another macro book that you create or rename the macro book before upgrading to prevent this loss.

OCLC macro book files and macro book files you create are stored in **X:\Program Files\OCLC\Connexion\Program\Macros** (X = the letter of your hard drive). See more about user file locations in *Getting Started with OCLC Connexion Client*.

To view the OCLC macros:

Action	
1	Click <b>Tools &gt; Macros &gt; Manage</b> , click the plus sign (+) next to OCLC to expand the list of OCLC macros.
2	In the Macro window under the <b>Macros</b> list, click the plus sign (+) next to OCLC to expand the list of macros OCLC supplies.
3	Click the name of a macro to see its description in the <b>Description</b> box, or double-click the name to open and view the macro in the Macro Editor.

## 2. Use Connexion client macros

### Client macro tools

The client provides the following tools for working with macros:

Macro tool:	To open, go to:
Manager	<p><b>Tools &gt; Macros &gt; Manage</b></p> <p>The Macro window opens. You can use it as a manager to create, edit, run, delete, and organize macros.</p>
Organizer	<p><b>Tools &gt; Macros &gt; Manage. Click Organizer</b></p> <p>The Macro Organizer window opens.</p> <p>Macros are stored in *.mbk files called <i>books</i>. Using the organizer, you can create or rename macro books, delete a macro or macro book, and move or copy macros among macro books.</p> <p><b>Caution</b>  <b>Prevent loss of data when moving or duplicating lengthier macros:</b> For moving or duplicating lengthy macros, OCLC recommends copying and pasting directly from one macro to another using the macro editor, rather than using the macro organizer.</p>
Recorder	<p><b>Tools &gt; Macros &gt; Manage. Click Record.</b></p> <p>The Record a New Macro window opens. After you enter a name and description for the new macro and click OK, the activities you do in the client are recorded as the new macro.</p>
Editor and debugger	<p><b>Tools &gt; Macros &gt; Manage. Click Edit.</b></p> <p>The OCLC Connexion Macro Editor and Debugger opens. You can use it to create, edit, and debug macros.</p> <p>The Macro Editor has its own Help files. To open:</p> <ul style="list-style-type: none"> <li>On the Macro Editor toolbar, click  (Help).</li> </ul> <p><b>Note:</b> This Help file does not open under Windows Vista or Windows 7 unless you download a Help engine file from the Microsoft Web site. See more information in "Introduction to OCLC Macro Language for Connexion client."</p>

## How to create macros

There are two ways to create macros for the client. You can:

- Write your own macros in the macro editor, or copy and paste from an existing macro to a new one and then edit the new macro as needed (see “Write macros” immediately following).

Or

- Record macros automatically using the macro recorder: You start the recorder, do a series of actions in a procedure you want to automate (using the mouse or the keyboard), and the recorder writes a macro that captures the procedure. See section titled “Record macros (create automatically)” below.

**Note:** See a list of client commands that are not available for use in macros you write or record below. The following commands are available for use in macros you write but cannot be recorded: View the next or previous 100 records from a list (**View > Previous 100 Records** or **Next 100 Records**).

## Write macros

Action	
1	On the Tools menu click <b>Macros &gt; Manage</b> , or press <Ctrl><Alt><Shift><G>.
2	In the Macro window under the <b>Macros</b> list, select the macro book in which you want to create the new macro. You must use macro books to contain macros. Or Create a new macro book in which to store the new macro: <ol style="list-style-type: none"> <li>1. In the Macro window box, click <b>New Book</b>.</li> <li>2. Type a descriptive name over the highlighted default name <i>New Book</i>.</li> <li>3. Press &lt;Enter&gt;.</li> </ol> You can also use the macro organizer to create a new macro book.
3	Click <b>New Macro</b> .
4	In the Macro Description window, type a unique description of the macro's purpose. Then click <b>OK</b> or press <Enter>.
5	In the text box with NewMacro highlighted, type an appropriate name for the new macro. Press <Enter>.
6	Click <b>Edit</b> .  The OCLC Connexion Macro Editor and Debugger opens. Its title bar gives the names of the macro book and macro. In the work area, the start and end text of a macro are provided:  <i>MacroName:[name you typed]</i> <i>'MacroDescription:[description you typed]</i>  <i>Sub Main()</i>  <i>End Sub</i>

Action	
7	<p>Between <i>Sub Main</i> and <i>End Sub</i>, type commands and appropriate comments to create the macro.</p> <ul style="list-style-type: none"> <li>To create a specific instance of the client object class, macros that use client commands must begin with the lines:  <code>Dim CS as Object</code>  <code>Set CS = CreateObject("Connex.Client")</code></li> <li>In macros that use variables (numbers, strings), include lines to declare the data type of each variable to be used in the macro.  <b>Example:</b>            The first part of a macro using the GetField command declares the variable <code>sData</code> as a string to store the text of the specified field:  <code>Dim CS as Object</code>  <code>Dim sData as String</code>   <code>Set CS = CreateObject("Connex.Client")</code></li> </ul>
8	Press <Enter> at the end of each line. Make sure <i>End Sub</i> remains as the last line of the macro.
9	On the Macro Editor toolbar, click  (Check) to check macro syntax. (See the "Check macro syntax" section below for details.)
10	Click  (Execute),  (Animate),  (Step Into), and  (Run to Cursor) to test and debug the macro. For more information, open the OML Macro Editor Help: Click  (Help) on the Macro Editor toolbar.
11	Click  (Save).
12	When finished editing, on the File menu click <b>Exit</b> , or press <Ctrl><F4>.  See a detailed macro example below.

## Tips for using the Macro Editor

- **Description.** Provide an informative description for each macro you create.
- **Comments.** Include comments in the macro to document what each part of the macro does. Comment lines begin with a single quotation mark (').

### Example:

```
' Get the first item from the first list
If CS.Search("WC", "gon,wi,th,w") > 0 Then CS.CloseList
End If
```

- **Open and work with multiple macros:**

- To open another Connexion client macro, with the Macro Editor open:

On the client Tools menu, click **Macros**. The Macros window reopens. Select another macro in the list or create a new one and select it, and then click **Edit**. The macro opens in a separate window.

- To help navigate among multiple open macros in the Editor:

On the Macro Editor Window menu, click **Cascade** or **Tile**, or make a different macro window current (bring it to the top and set focus) by selecting its name in the windows list.

- **Help for BASIC keywords.** To get a specific Help topic on any BASIC language keyword (such as *sub*) in a macro in the Editor, right-click the keyword in the macro text.

**Reference table:** For diacritics and special characters in data used with macro commands such as *CS.GetField* or *CS.SetField*, see a list, “ASCII and hexadecimal codes for diacritics and special characters” in “Connexion client macro commands: Edit records.”

## View or edit macros

Action	
1	In the Macro window ( <b>Tools &gt; Macros &gt; Manage</b> ) under the <b>Macros</b> list, click the plus sign (+) next to the macro book that contains the macro you want to edit.
2	Select the macro and click Edit.
3	In the OCLC Connexion Macro Editor and Debugger window, edit the macro, and then follow steps 9 through 11 of the procedure above for creating a macro.

## Check macro syntax

When you create or edit a macro, you must check the macro syntax (click the **Check**  button on the toolbar). The Macro Editor displays a message on the status bar of the Editor window.

- If the macro contains no syntax errors, the message is: *Successfully compiled [macrobook]![macroname].sbl*
- If the macro contains syntax errors, the message is: *\_Error(s) in [macrobook]![macroname].sbl*. Lines containing syntax errors are highlighted in red.

To review errors:

Action	
	On the Macro Editor toolbar, click the <b>Next Error</b> button  and <b>Previous Error</b> button  to move between error liens in the macro.

To view details of syntax errors:

Action	
1	On the Macro Editor toolbar, click the <b>Console</b> button  .  A new window opens at the bottom of the Editor window, just above the gray message bar. The window displays output from the syntax checker.
2	Drag the separator up to expand the console window. Maximize the Editor window to provide more space if needed.
3	On the Macro Editor toolbar, click the <b>Check</b> button <input checked="" type="checkbox"/> again. In the console window, the Editor displays messages that give the line number containing each syntax error (in parentheses following the macro name) and identify the general type of error.

## Macro example

*Sub Main()*

```
Dim CS as Object
Set CS = CreateObject("Connex.Client")

If CS.Logon("", "", "") = True Then
    MsgBox "Logon successful"
End If

CS.Search "WC", "#1"

CS.AddFieldLine "300Line 10 test", 10
If CS.AddField(2, "500 Another test") = True Then
    MsgBox "500 field successfull added"
End If
```

*End Sub*

This example:

- Logs on to Connexion if not already logged on, using the default authorization and password you selected in **Tools > Options > Authorizations** tab (leaves logon values blank).
- Searches WorldCat for record number 1.
- Adds a new field 300 in the tenth line of the record (inserted as the tenth variable field) with no indicators (two spaces between the tag number 300 and the text of the field). Adds the text *Line 10 test* to the field. (No confirmation message.)
- Adds a second instance of repeatable field 500 (that is, adds a new field 500 below the first existing field 500) with no indicators and the text *Another test*. If the action is successful, returns the confirmation message *500 field successfully added*.

See a complete list of client commands in "List of Connexion client macro commands." See detailed descriptions and examples for each command in sections organized by function.

## Include macros in another macro

Including macros in a macro lets you:

- Reuse macros you developed and tested as building blocks for more complex macros
- Insert standardized elements in multiple macros (for example, declarations of variables, frequently used subroutines, or a function library)
- Avoid the errors and repetitive debugging that result from recreating existing code

You can:

- Run a macro from another macro:

- The *RunMacro* command executes a specified macro from within another macro. When the called macro finishes running, execution of the calling macro resumes with the next statement
- **Syntax:** *BOOL = CS.RunMacro("macrobook!macroname")*

**Tip:** For faster execution, locate macro(s) you run using the *RunMacro* command in the same macro book that contains the calling macro(s).

**Limitation:** *RunMacro* does not include code from the specified macro in the compile process. The called macro is treated as an external "black box."

- Include code from another macro in compile:
  - The *\$Include* metacommand tells the OML compiler to merge code from another macro at compile time. Included code becomes part of the macro that contains the *\$Include* command. Functions, subroutines, and variables in the included code become available for use at any time during the main macro's execution.
- **Syntax:** *\$Include: "macrobook!macroname"*
- **Correct OML syntax:** The topic describing *\$Include* in OML Standard Basic Language Reference Help gives the syntax for including a file by specifying path (drive and folder) and file name. This syntax does not work in client macros. You must specify the macro to include as *"macrobook!macroname"*.
- **Tip:** Place the *\$Include* metacommand before the main procedure (*Sub Main() ... End Sub*) of the macro if the code to be included contains global components such as declared variables, functions, or subroutines.

## Run macros

Action	
1	In the Macros list of the Macro window ( <b>Tools &gt; Macros &gt; Manage</b> ), click the plus sign (+) next to the macro book that contains the macro you want to run.
2	Select the macro.
3	Click <b>Run</b> .

**Tip:** You can also run a macro by assigning and using a keystroke shortcut or by assigning a user tool button and adding it to the toolbar. See procedures in "Customize keystroke shortcuts for menu items, macros, or characters" and "Assign custom user tools" in Basic/Set Options and Customize.

## Stop macros

Action
To stop a macro from running, on the Tools menu, click <b>Macros &gt; Stop</b> , or press <Ctrl><Alt><Shift><T>.

## Delete macros

Action	
1	In the Macro window ( <b>Tools &gt; Macros &gt; Manage</b> ) under the <b>Macros</b> list, click the plus sign (+) next to the macro book that contains the macro you want to delete.
2	Select the macro you want to delete.
3	Click <b>Delete</b> . Click <b>Yes</b> to confirm or <b>No</b> to cancel.

## Organize macros and macro books

- You create macros only within a macro book. Macro book files use the extension \*.mbk

Action	
1	In the Macro window ( <b>Tools &gt; Macros &gt; Manage</b> ), click <b>Organizer</b> .
2	<p>In the Macro Organizer window, you can:</p> <ul style="list-style-type: none"> <li>Create a macro book: Click <b>New Macro Book</b> and follow steps 4 and 5 in the procedure above for creating macros.</li> <li>Rename a macro book: Select a macro book. Click <b>Rename Macro Book</b>, and type a new name over the old name.</li> <li>Delete a macro book: Select a macro book. Click <b>Delete</b>. Click <b>Yes</b> to confirm or <b>No</b> to cancel.</li> <li>Move or copy a macro book: <ul style="list-style-type: none"> <li>To move or copy a macro book to or from another location, change the location for <b>Macro Directory 1</b> and/or <b>Macro Directory 2</b>. Click the arrow to open the <b>Current</b> list. Select the drive and then the folder to move or copy to or from.</li> </ul> <p><b>Default location:</b>  <b>X:\Program Files\OCLC\Connexion\Program\Macros\</b>, where <b>X</b> is the letter of your hard drive</p> <ul style="list-style-type: none"> <li>Select the macro book you want to move or copy and click <b>Move</b> or <b>Copy</b>.</li> </ul> <p>The highlighted macro book is moved or copied from its current location to the location shown under Current Directory in the opposite list area</p> </li> </ul> <ul style="list-style-type: none"> <li>Move or copy a macro between macro books: <ul style="list-style-type: none"> <li>To move or copy a macro book to or from another location, change the location for <b>Macro Directory 1</b> and/or <b>Macro Directory 2</b>. Click the arrow to open the Current list. Select the drive and then the folder to move or copy to or from.</li> <li>In either list, select the macro book <b>to</b> which you want to move or copy a macro.</li> <li>In the other list, select the macro book <b>from</b> which you want to move or copy a macro. Click the plus sign (+) to list the macros in the macro book.</li> <li>Select the macro and click <b>Move</b> or <b>Copy</b>.</li> </ul> <p><b>Caution for moving or duplicating lengthy macros:</b> To prevent loss of data, OCLC recommends copying and pasting directly from one macro to another using the macro editor rather than using Move or Copy in the macro organizer.</p> </li> </ul>

## Caution: OCLC-supplied macro books may be overwritten at upgrade

### Do not store customized macros in OCLC-supplied macro books

OCLC-supplied macro book files **OCLC.mbk** and **Dewey.mbk** may be overwritten when you upgrade the software to provide you with the most currently available OCLC macros.

If you store customized macros in either of these books, you must rename the existing macro book before upgrading to be sure that the installation does not overwrite it and cause you to lose your own macros.

**OCLC recommends that you:** Create different macro book file(s) to store macros you create, copy, or record.

## Actions not available for macros

### You cannot use the following actions in macros you write or record:

- Attach digital files to WorldCat records (Edit menu) (available only to those who have an OCLC logon authorization that includes CONTENTdm, a license to use CONTENTdm, and an established CONTENTdm collection and who use the CONTENTdm hosting service)
- Authorities history records searching and browsing (Authorities > Search menu) and showing distributed vs. superseded versions (Authorities > Show menu)
- Batch set or delete holdings by OCLC number (Batch menu)
- Cancel the last edit in a record (Edit > Undo)
- Check spelling (Tools menu)
- Control a single heading in a record (Edit > Control Headings > Single)
- Convert characters to MARC-8 CJK (Edit > MARC-8 Characters menu)
- Copy data as text (Edit > Copy as Text)
- Copy the OCLC control number by selecting it manually and copying (to record, you must use the Copy Control Number command on the Edit menu)
- Create constant data (Cataloging > Create > Constant Data)
- Create records for electronic resources by extracting metadata (Cataloging > Create > Extract Metadata)
- Customize the toolbar (Tools > Toolbar Editor)
- Delete selected text
- Derive a new constant data record (Edit > Derive > New Constant Data)
- Edit or create 006, 541, or 583 fields using guided entry (Edit > Guided Entry > Edit [or Insert 006, 541, 583])

**Note:** Existing macro commands can be used, such as SetFieldLine to insert a field and specify data.

- Import records (File > Import Records)
- Insert field 760 - 787 data automatically from a cited record (Edit > Insert from Cited Record)
- Launch local holdings maintenance (Tools menu)
- Launch WebDewey (Cataloging menu)
- Link/unlink non-Latin script fields with equivalent Latin script fields (Edit > Linking Fields)
- Lock linked authority record (Action menu)

- Move the cursor from one record area to another (Edit > Previous Record Area or Next Record Area)
- Open the CJK E-Dictionary (Tools menu)
- Paste Unformatted (Edit menu)
- Print accessions lists (File menu)
- Set a constant data record as the default (Edit > Constant Data > Online > Set as Default)
- Set options in Tools > Options

**Note:** You can, however, assign keystroke shortcuts to run macros, either by writing a macro using the command AssignMacroToKey or recording a macro while you assign a keystroke from Tools > Keymaps.

- Show default constant data (Cataloging > Show > Default Constant Data; same command on the Authorities menu)
- Show numbers of online save file records by individual statuses (Cataloging > Show > By Online Save File Status; same command on the Authorities menu)
- Transliterate Arabic or Persian (Edit menu)
- Use assigned User Tools (Tools menu)
- Use text strings (Tools menu)
- Verify MARC-8 characters (Edit menu)

**Note:** You can use the following commands in macros you write, but you cannot record them:

- GetNext100Records (View menu)
- GetPrevious100Records (View menu)

### 3. Record macros (create automatically)

#### Record a macro

The client provides a macro recorder that records the actions in a procedure as you do them, whether you use the mouse to click commands or toolbar buttons, or press shortcut keystrokes, or use a combination of mouse and keyboard.

The recorder writes the macro automatically in the OML macro recorder.

To record a macro:

Action	
1	On the Tools menu, click <b>Manage</b> , or press <Ctrl><Alt><Shift><G>.
2	In the Macro window, click the name of the macro book where you want to store the new macro and then click <b>Record</b> .
3	In the Record a New Macro window, enter a unique name and description for the macro. The description displays in: <ul style="list-style-type: none"> <li>The Description box of the Macro window, when you select the name of the macro, to help you remember what the macro does</li> <li>At the top of the macro text when you open the macro in the macro editor</li> </ul>
4	Click <b>OK</b> to start recording, or click <b>Cancel</b> to close the window without starting recording. <p><b>Result</b> if you click <b>OK</b>:</p> <ul style="list-style-type: none"> <li>The client displays a macro recording toolbar under the client toolbar:  </li> <li>The toolbar contains buttons for (left to right): Pause Recording, Resume Recording, and Stop Recording.</li> <li>These commands are also available as commands under <b>Tools &gt; Macros</b>.</li> </ul>
5	Carry out the procedure(s) you want to capture in a macro. The client records all actions except those listed in “Non-recordable actions” below.
6	Optional: To pause and then resume recording the current new macro: <ul style="list-style-type: none"> <li>On the Tools menu, click <b>Macros &gt; Pause Recording</b>.</li> <li>Or</li> <li>On the macro recording toolbar, click .</li> <li>Or</li> <li>Press &lt;Alt&gt;&lt;T&gt;&lt;M&gt;&lt;P&gt;.</li> </ul> <p>To resume recording the same macro at the point where you paused:</p> <ul style="list-style-type: none"> <li>On the Tools menu, click <b>Macros &gt; Resume Recording</b>.</li> <li>Or</li> <li>On the macro recording toolbar, click .</li> <li>Or</li> <li>Press &lt;Alt&gt;&lt;T&gt;&lt;M&gt;&lt;E&gt;.</li> </ul>

Action	
7	<p>When you are finished, to stop recording and end the macro: On the Tools menu, click <b>Macros &gt; Stop Recording</b></p> <p>Or</p> <p>On the macro recording toolbar, click .</p> <p>Or</p> <p>Press &lt;Alt&gt;&lt;T&gt;&lt;M&gt;&lt;R&gt;.</p> <p><b>Results:</b></p> <ul style="list-style-type: none"> <li>• The macro is saved automatically.</li> <li>• You can open the macro to check or edit it. In the Macro window (<b>Tools &gt; Options &gt; Macros &gt; Manage</b>), click the name of the macro you recorded, and then click <b>Edit</b>. The macro opens in the Macro Editor.</li> <li>• The macro recording toolbar closes.</li> </ul> <p><b>Note:</b> If you exit the client before you stop recording a macro, the program stops recording for you. You may want to check the macro later for completeness.</p>
8	<p>To view, check, or edit the macro, see “View or edit macros” and other sections about checking and editing macros in the topic, “Use Connexion client macros.”</p>

### Tips for using the macro recorder

- To copy from field 245, use Find/Replace to find field 245. Then close the box to locate the cursor in field 245.
- You cannot record manually selecting and copying the OCLC number in a record. You must use **Edit > Copy Control Number**.
- When recording taking an action on a bibliographic record, the macro recorder does not record replacing text by manually selecting and overwriting the text.

## 4. List of Connexion client macro commands

This section has two lists of client macros:

- Commands listed by function
- Commands listed in alphabetical order

### List of client macro commands by function

Log on and off	
Command name	Brief description
Logon	Logs on to OCLC Connexion
Logoff	Logs off OCLC Connexion
IsOnline	Checks whether you are currently logged on

Search	
Command name	Brief description
Search	<p>Lets you use a search string to search an online database</p> <ul style="list-style-type: none"> <li>• WorldCat</li> <li>• Online bibliographic save file</li> <li>• Online bibliographic constant data</li> <li>• Local bibliographic save file</li> <li>• Local bibliographic constant data</li> <li>• Online bibliographic review records</li> <li>• Authority File</li> <li>• Online authority save file</li> <li>• Online authority constant data</li> <li>• Local authority save file</li> <li>• Local authority constant data</li> <li>• Online authority review records</li> </ul>
Browse	Lets you browse WorldCat for titles or browse the Authority File for headings
BrowseAuthorityExpanded	Lets you expand a browse for a heading by specifying a subheading

Edit records	
Command name	Brief description
AddField	Adds a field in correct tag order and specifies the order of its appearance among other fields with the same tag number
AddFieldLine	Adds a variable field in a specific field position (line number)
CancelChanges	Cancels all editing changes made to a master record (if from WorldCat or the Authority File) or all changes made since you last saved the record (if from the save file)
ControlHeadingsAll	Controls all controllable headings in a record that are not already controlled

<b>Edit records (cont.)</b>	
<b>Command name</b>	<b>Brief description</b>
UncontrolHeading	Uncontrols a controlled heading where the cursor is located so that you can edit the heading
UncontrolAll	Uncontrols all controlled headings in the current record
IsHeadingControlled	Determines whether a heading in a specified line of the current record is controlled
Copy	Copies specific text and places it on the clipboard
Cut	Cuts specific text and places it on the clipboard
Paste	Pastes text from the clipboard at a specified insertion point
CopySelected	Copies selected text and places it on the clipboard
CutSelected	Cuts selected text and places it on the clipboard
CopyField	Copies the field where the cursor is located and places it on the clipboard
CopyControlNumber	Copies the OCLC control number (bibliographic records) or ARN (authority records) and places it on the clipboard
CursorPosition	Indicates or changes the position of the cursor in a record, where the cursor position is variable, depending on the presence of data (for use by the macro recorder; use CursorColumn when writing macros)
DeleteCurrentField	Deletes the field where the cursor is located
DeleteCurrentWord	Deletes the word where the cursor is located
DeleteToEndOfCell	Deletes text from the cursor position to the end of the cell in a field
MoveFieldUp	Move the current field above the preceding field
MoveFieldDown	Move the current field below the following field
DeleteField	Deletes a field by specific tag number and, in the case of repeated variable fields, by order of appearance among the other fields with the same tag
DeleteFieldLine	Deletes a field in a specific field position (line number)
GetField	Gets the data for a specific field by tag number and, in the case of repeated variable fields, by order of appearance among the other fields with the same tag
GetFieldLine	Gets the data for a field in a specific field position (line number)
GetFieldUnicode	Works like GetField, but also returns Unicode data in Numeric Character Reference (NCR) form
GetFieldLineUnicode	Works like GetFieldLine, but also returns Unicode data in Numeric Character Reference (NCR) form
SetField	Adds data to a field specified by tag number and, in the case of repeated variable fields, by order of appearance among the other fields with the same tag
SetFieldLine	Adds data to a field in a specific field position (line number)
SetFixedField	Inserts a value in a specific fixed-field element
GetFixedField	Gets the data for a specific fixed-field element
FixedFieldPosition	Returns or specifies how the fixed field displays in a record
SetCursorFixedField	Places the cursor and selects the value, if any, in a specified element of the fixed field
GetSelectedText	Gets selected text from a record (text can span multiple fields)
SetSelectedText	Inserts specific text in a record at the cursor location (text can span multiple fields)

Edit records (cont.)	
Command name	Brief description
InsertText	Inserts specified text at the cursor location
FindText	Finds and highlights the next occurrence of specified text
ReplaceText	Replaces the highlighted “found” text with specified text
ReplaceTextAll	Replaces all occurrences of “found” text with specified text
FindReplaceNextText	Finds the next occurrence of the specified text and selects it. Another instance of FindReplaceNextText replaces the selected text, moves to the next occurrence, and selects it (combines FindText and ReplaceText commands)
Reformat	Reformats a record (redispays the record with fields in correct order)
Validate	Validates a record or records selected in a list and returns a list of validation errors, if any (logged on only)
CursorColumn	Indicates or changes the column where the cursor is located in a record
CursorRow	Indicates or changes the row where the cursor is located in a record
InsertMode	Indicates or changes the insert mode (insert mode on or off)
ApplyOnlineDefaultCD	Applies your online default constant data record to the current record or to records selected in a list
ApplyOnlineCDByName	Applies an online constant data record by its specified name to the current record or to records selected in a list
ApplyCDFromList	Specifies a constant data record on a list and applies it to the current record or to records selected in a list <b>Note:</b> Before version 1.20 of the client, this macro command was named <i>ApplyOnlineCDFromList</i> . Both macro names work with both online and local constant data lists.
ApplyLocalDefaultCD	Applies your local default constant data record to the current record or to records selected in a list
ApplyLocalCDByName	Applies a local constant data record by its specified name to the current record or to records selected in a list
ChangeRecordType	Changes the MARC format of a record (for example, from Books format to Computer Files format)
InsertFromCitedRecord	Inserts data from a cited record that has a bibliographic connection to the record you are editing.
GetClassNumber	Retrieves a classification number for the displayed record using the specified classification scheme (Dewey or Library of Congress). Gets a suggested call number from the prototype OCLC Classify Web service ( <a href="http://classify.oclc.org/">http://classify.oclc.org/</a> ) by matching records based on the following standard numbers, in this order: OCLC number, ISBN, ISSN, UPC, LCCN.

See also “ASCII and hexadecimal codes for diacritics and special characters.”

**Note on macros and non-Latin script data:** You can incorporate Unicode character data into macros using GetFieldUnicode and GetFieldLineUnicode. The Unicode characters are converted to Numeric Character Reference (NCR) form.

<b>Create records</b>	
<b>Command name</b>	<b>Brief description</b>
OpenWorkform	<p>Opens a bibliographic workform of a specified type (Books, Computer Files, Continuing Resources*, Maps, Mixed Materials, Scores, Sound Recordings, or Visual Materials)</p> <p>Or</p> <p>Opens an authority workform of a specified type (Blank Record, Conference Name, Corporate Name, Geographics, Personal Name, Series, Uniform Title)</p> <p>*Formerly called Serials; includes serials and integrating resources material types; see description for more information.</p>
DeriveNewRecord	Derive a new record or a new constant data record from an existing record
OpenWorkformForDatabase	Alternative to OpenWorkform command

<b>Take actions on records</b>	
<b>Command name</b>	<b>Brief description</b>
UpdateHoldings	<p>Updates holdings for a record or for records selected in a list (adds your OCLC institution symbol)</p> <p>Or</p> <p>When offline, marks local record(s) for batch update</p>
ProduceandUpdateHoldings	<p>Produces cards and accessions lists offline, as specified in your OCLC cataloging profile, and updates holdings for a record or for records selected in a list</p> <p>Or</p> <p>When offline, marks local record(s) for batch update and produce</p>
AltProduceandUpdateHoldings	<p>Updates holdings (adds OCLC symbol) and specifies Ad Produce, All Produce, and/or the number of extra cards to produce for a record or for records selected in a list</p> <p>Or</p> <p>When offline, marks local record(s) for batch update and alternate produce</p>
DeleteHoldings	<p>Deletes your holdings (OCLC symbol) from a record or from records selected in a list</p> <p>Or</p> <p>When offline, marks local record(s) for batch delete holdings</p>
Lock	Locks a record
ReleaseRecordLock	Releases the lock on a record
ReplaceRecord	<p>Replaces a record or records selected in a list in WorldCat or the Authority File</p> <p>Or</p> <p>When offline, marks local record(s) for batch replace</p>

Take actions on records (cont.)	
Command name	Brief description
ReplaceandUpdateHoldings	Replaces a record, or records selected in a list, and updates holdings Or When offline, marks local record(s) for batch replace and update
SaveOnline	Saves a record, or records selected in a list, to the online save file
SaveToLocalFile	Saves a record, or records selected in a list, to the local save file
DeleteRecord	Deletes a record, or for records selected in a list, from the online save file
Export	Exports a record or records selected in a list, while logged on or offline Or Marks record(s) for batch export if you set the option in <b>Tools &gt; Options &gt; Batch</b>
AddToAuthorityFile	Adds a new record or records selected in a list, to the Authority File (NACO-authorized libraries only) Or When offline, marks local record(s) for batch add
SubmitForReview	Sends a bibliographic or authority record, or records selected in a list, to another library for review Or When offline, marks local record(s) for batch submit
RecallSubmitted	Withdraws a record you submitted for review
AddToOnlineDatabase	Alternative to AddToAuthorityFile command

Work with records, lists, and files	
Command name	Brief description
SearchList	Searches for specified data in a list
GetListItem	Display a specified record on a list
GetFirstSelectedItem	Opens the first selected entry in the current list
GetNextSelectedItem	Opens the next selected entry in the current list
GetFirstItem	Selects the first item in a list (can be another list or record)
GetNextItem	Selects the next item in a list
GetPrevItem	Selects the previous item in a list
GetListCellData	Selects data or value in a cell specified by row and column title
GetListCellDataUnicode	Selects data or value that contains non-Latin scripts in a cell specified by row and column title
GetNext100Records	Displays a list of the next 100 records retrieved from the Authority File or from the bibliographic or authority online save file or online constant data
GetPrevious100Records	Displays a list of the previous 100 records retrieved from the Authority File or from the bibliographic or authority online save file or online constant data
CloseList	Closes a list
CloseRecord	Closes a record
GetWindowTitle	Returns the specified window title

<b>Work with records, lists, and files (cont.)</b>	
<b>Command name</b>	<b>Brief description</b>
ItemType	Distinguishes the type of record or list for the current window
SetTopWindow	Makes a specified window the current window (top-most or active window)
WindowCount	Returns the total number of windows open in the client
CreateLocalFile	Creates a local file in a specified path or sets an existing local file as the default
CopyRecord	Copies a local file record or records selected in a local file list to another, existing local file that you specify
MoveRecord	Moves a local file record or records selected in a local file list to another, existing local file that you specify.
Macro commands for navigating in records	<p>The following commands move the cursor to a new location. Most correspond to keys on the keyboard. They are used primarily for the macro recorder.</p> <p><b>Macro command: Moves the cursor to...</b></p> <ul style="list-style-type: none"> <li>• EndCell: To the end of the current cell</li> <li>• HomeCell: To the beginning of the current cell</li> <li>• EndRecord: To the end of the variable fields</li> <li>• HomeRecord: To the beginning of the variable fields</li> <li>• KeyUp: One line up</li> <li>• KeyDown: One line down</li> <li>• KeyRight: One position to the right</li> <li>• KeyLeft: One position to the left</li> <li>• NextSubfield: To the next subfield</li> <li>• PrevSubfield: To the previous subfield</li> <li>• DeleteKey: One position to the right and deletes the character or space to the right of the cursor's original position</li> <li>• Backspace: One position to the left and deletes the character or space to the left of the cursor's original position</li> </ul>

<b>Print and work with macros, statuses, holdings, labels, and profiles</b>	
<b>Command name</b>	<b>Brief description</b>
SetMyStatus	Adds or changes My Status for a record or for records selected in a list
SetWorkflowStatus	Changes the workflow status for a record or for records selected in a list
QueryRecordStatus	Returns a specific status for a record
IsHeldBy	Determines whether a specific OCLC symbol is included in a record
Holdings	Returns a record's holdings list
Print	Prints a record or (available for local save file and local constant data lists only) prints records selected in a list
PrintList	Prints a list or a list of selected entries
PrintLabel	Prints labels based on field data in the record(s) for spine, call number, author, and title
PrintLabelEx	Allows you to edit data for a label for the current record before printing the record

Print and work with macros, statuses, holdings, labels, and profiles (etc.)	
Command name	Brief description
GetLabelSpine	Returns the spine text extracted from the current record, based on the holding library code, so that you can edit it
GetLabelSpineUnformatted	Returns unformatted text (no line breaks) for a spine label extracted from the current records, based on the holding library code, so that you can edit it
GetLabelCallNumber	Returns the call number from the current record, based on the holding library code, so that you can edit it
GetLabelAuthor	Returns the author text from the current record, so that you can edit it
GetLabelTitle	Returns the title text from the current record, based on the holding library code, so that you can edit it
StartLabelBatch	Starts the label print process (must precede PrintLabel or PrintLabelEx)
EndLabelBatch	Ends the label print process (must precede PrintLabel or PrintLabelEx)
PrintMacro	Prints a macro
RunMacro	Runs a macro
AssignMacroToKey	Assigns a specified keystroke shortcut to a macro
LoadProfile	Loads a specified user profile (for those who use multiple profiles)
SendKeys	Sends keystrokes to the client

## Alphabetical list of macro commands

A – F	G – O	P - Z
AddField	GetClassNumber	Paste
AddFieldLine	GetField	PrevSubfield
AddToAuthorityFile	GetFieldLine	Print
AddToOnlineDatabase	GetFieldLineUnicode	PrintLabel
AltProduceAndUpdateHoldings	GetFieldUnicode	PrintLabelEx
ApplyCDFFromList	GetFirstItem	PrintList
ApplyLocalCDBByName	GetFirstSelectedItem	PrintMacro
ApplyLocalDefaultCD	GetFixedField	ProduceAndUpdateHoldings
AssignMacroToKey	GetLabelAuthor	QueryRecordStatus
Backspace	GetLabelCallNumber	RecallSubmitted
Browse	GetLabelSpine	Reformat
BrowseAuthorityExpanded	GetLabelSpineUnformatted	ReleaseRecordLock
CancelChanges	GetLabelTitle	ReplaceAndUpdateHoldings
ChangeRecordType	GetListCellData	Repl
CloseList	GetListCellDataUnicode	ReplaceRecord
CloseRecord	GetListItem	ReplaceText
ControlHeadingsAll	GetNext100Records	ReplaceTextAll
Copy	GetNextItem	RunMacro
CopyControlNumber	GetNextSelectedItem	SaveOnline
CopyField	GetPrevious100Records	SaveToLocalFile
CopyRecord	GetPrevItem	Search
CopySelected	GetSelectedText	SearchList
CreateLocalFile	GetWindowTitle	SendKeys

A – F	G – O	P - Z
CursorColumn CursorPosition CursorRow Cut CutSelected DeleteCurrentField DeleteCurrentWord DeleteField DeleteFieldLine DeleteHoldings DeleteKey DeriveNewRecord EndCell EndLabelBatch EndRecord Export ExportAppend FindReplaceNextText FindText FixedFieldPosition	Holdings HomeCell HomeRecord InsertFromCitedRecord InsertMode InsertText IsHeadingControlled IsHeldBy IsOnline ItemType KeyDown KeyLeft KeyRight KeyUp LoadProfile Lock Logoff Logon MoveFieldDown MoveFieldUp MoveRecord NextSubfield OpenWorkform OpenWorkformForDatabase	SetCursorFixedField SetField SetFieldLine SetFixedField SetMyStatus SetSelectedText SetTopWindow SetWorkflowStatus StartLabelBatch SubmitForReview UncontrolAll UncontrolHeadings UpdateHoldings Validate WindowCount

## 5. Connexion client macro commands: Log on and off

### Log on

<b>Syntax</b>	BOOL= CS.Logon (sAuthorization, sPassword, sLevel)
<b>Use to</b>	Log on to OCLC Connexion.  Same as using the <b>File &gt; Log On</b> menu command, except that the Message of the Day is not displayed after logon is completed.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sAuthorization</b>, enter your OCLC authorization number (in the format 123-456-789 or 123456789) enclosed in quotation marks (“”).</li> <li>For <b>sPassword</b>, enter your OCLC password enclosed in quotations marks (“”).</li> </ul> <p><b>Note:</b> Leave sAuthorization and sPassword blank to use your default authorization and password as selected in <b>Tools &gt; Options &gt; Authorizations tab</b>. Enter only a set of quotation marks for each parameter.</p> <ul style="list-style-type: none"> <li>For <b>sLevel</b>, leave blank. Enter only a set of quotation marks. This parameter may be implemented in the future.</li> </ul>
<b>Return values</b>	TRUE if logon is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()   Dim CS As Object   Set CS = CreateObject("Connex.Client")   If CS.Logon("", "", "") = True Then     MsgBox "Logon successful"   End If End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>Logs on using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b></li> <li>If logon is successful, returns the message <i>Logon successful</i>.</li> </ul>

### Log off

<b>Syntax</b>	CS.Logoff
<b>Use to</b>	Log off OCLC Connexion.  Same as using the <b>File &gt; Log Off</b> menu command
<b>Parameters</b>	None
<b>Return values</b>	No value
<b>Example</b>	<pre>Sub Main()   Dim CS As Object   Set CS = CreateObject("Connex.Client")   CS.Logoff End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>Logs off from the system (ends your session).</li> </ul>

## IsOnline

<b>Syntax</b>	BOOL = CS.IsOnline
<b>Use to</b>	Determine whether you are logged on.
<b>Parameters</b>	None
<b>Return values</b>	True if logged on; otherwise, FALSE.
<b>Example</b>	<pre>Sub Main()     Dim CS As Object     Set CS = CreateObject("Connex.Client")     If CS.IsOnline = True Then         MsgBox "You are currently logged on"     Else         MsgBox "You are not currently logged on"     End If End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"><li>• If logged on, returns the message <i>You are currently logged on</i></li><li>• Or</li><li>• If not logged on, returns the message <i>You are not currently logged on.</i></li></ul>

## 6. Connexion client macro commands: Search

### Search

<b>Syntax</b>	nNumRecords = CS.Search (sDatabase, sSearchTerm)
<b>Use to</b>	<p>Search an online database using a search string.</p> <p>Same as using the following menu commands:</p> <ul style="list-style-type: none"> <li>• Cataloging &gt; Search &gt; WorldCat</li> <li>• Cataloging &gt; Search &gt; Online Save File</li> <li>• Cataloging &gt; Search &gt; Online Constant Data</li> <li>• Cataloging &gt; Search &gt; Local Save File</li> <li>• Cataloging &gt; Search &gt; Local Constant Data</li> <li>• Authorities &gt; Search &gt; Authority File</li> <li>• Authorities &gt; Search &gt; Online Save File</li> <li>• Authorities &gt; Search &gt; Online Constant Data</li> <li>• Authorities &gt; Search &gt; Local Save File</li> <li>• Authorities &gt; Search &gt; Local Constant Data</li> </ul>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• <b>nNumRecords</b> represents the return value: the number of records found.</li> <li>• For <b>sDatabase</b>, enter one of the following database identifiers, enclosed in quotation marks (""): <ul style="list-style-type: none"> <li>○ <b>WC</b> (WorldCat)</li> <li>○ <b>BS</b> (online bibliographic save file)</li> <li>○ <b>BD</b> (online bibliographic constant data)</li> <li>○ <b>BL</b> (local bibliographic save file)</li> <li>○ <b>BC</b> (local bibliographic constant data)</li> <li>○ <b>BR</b> (online bibliographic review records)</li> <li>○ <b>AF</b> (Authority File)</li> <li>○ <b>AS</b> (online authority save file)</li> <li>○ <b>AD</b> (online authority constant data)</li> <li>○ <b>AL</b> (local authority save file)</li> <li>○ <b>AC</b> (local authority constant data)</li> <li>○ <b>AR</b> (online authority review records)</li> </ul> </li> </ul> <p><b>Note:</b> If the database parameter is empty or the database identifier is not supported, the search defaults to WorldCat.</p> <ul style="list-style-type: none"> <li>• For <b>sSearchTerm</b> enter a search string (in upper- or lowercase) that is valid for the database you want to search. Enclose in quotation marks ("").  <b>Note:</b> Whole phrase WorldCat searches require that the phrase be enclosed in quotation marks. For these searches, enclose the phrase in a double set of quotation marks ("search phrase").</li> </ul> <p><b>Tip: Retrieve all records from save file or constant data:</b> To retrieve all records from an online save file or from online constant data, specify the database identifier and leave the search string empty.</p> <p><b>Example:</b>  <b>CS.Search "BS", ""</b> (retrieves all records in the online bibliographic save file)</p>
<b>Return values</b>	Number of records found (nNumRecords)
<b>Example</b>	<pre>Sub Main()   Dim CS As Object   Set CS = CreateObject("Connex.Client")</pre>



- In a keyword search, you can specify proximity and order of terms with proximity operators **with (w)** or **near (n)**.
- To expand a term in a keyword or numeric search, replace characters or endings with symbols for wildcards (# or ?), truncation (\*), and plural stemming (+). See Use wildcards, truncation, and plural stemming in searches for details.

**Caution:** Wildcard and truncation symbols must be preceded by at least three characters.

For more information and for lists of selected indexes and examples, see Cataloging, Search WorldCat, “Search WorldCat interactively.”

See [Searching WorldCat Indexes](#) for comprehensive descriptions of indexes, guidelines, and examples.

**Online bibliographic save file searches (database identifier BS)**

Guidelines:

- Enter an index label (see list below), a colon, and then the search term.
- You can combine searches using Boolean operators. To distinguish the words **and**, **or**, or **not** when they are part of your search term (not used as Boolean operators), you must enclose in single quotation marks (').

**Example:**

**“ti:red 'and' the black”**

- The search is not case-sensitive: you can enter the search string in upper- or lowercase.
- To retrieve all records in the save file, leave the search string parameter empty: **CS.Search “BS”, “”**.

Indexes and labels:

Index	Label	Values
Save File Number	<b>sf:</b> (default)	You supply
OCLC Control Number	<b>an:</b>	You supply
Title	<b>ti:</b>	You supply
URL	<b>am:</b>	You supply
Used Date	<b>us:</b>	You supply
My Status	<b>ms:</b>	You supply
Action Status	<b>as:</b>	<b>delete holdings completed</b> <b>delete holdings failed</b> <b>export completed</b> <b>export failed</b> <b>export ready</b> <b>label completed</b> <b>label failed</b> <b>label ready</b> <b>produce completed</b> <b>produce failed</b> <b>replace completed</b> <b>replace failed</b> <b>update holdings completed</b>

Index	Label	Values
-------	-------	--------

Action Status (cont.)	<b>as:</b>	<b>update holdings failed</b> <b>validate completed</b> <b>validate failed</b>
Workflow Status	<b>ws:</b>	<b>completed</b> <b>in process</b> <b>in review</b> <b>new</b>
Source Status	<b>ss:</b>	<b>derived</b> <b>extracted</b> <b>imported</b> <b>oclc</b> <b>workform</b>
Review	<b>rs:</b>	<b>submitted</b>

Note: If you record rather than write a macro, statuses are recorded with no spaces between words. When you write macros, you can retain the spaces. Values you enter are not case-sensitive.

### Online bibliographic constant data searches (database identifier **BD**)

Search examples:

**CS.Search "BD", "nm:bibconstantdata"** (retrieves a constant data record named *bibconstantdata*)

**CS.Search "BD", ""** (retrieves all bibliographic constant data records)

Indexes and labels:

- Constant Data Name (**nm**) (default)
- My Status (**ms**)

### Local bibliographic save file searches (database identifier **BL**)

Follow the same guidelines as for online bibliographic save file searches (except use the database identifier **BL**).

Indexes and labels:

Index	Label	Values
Save File Number	<b>sf:</b> (default)	You supply
OCLC Control Number	<b>an:</b>	You supply
Title	<b>ti:</b>	You supply
Name	<b>nm:</b>	You supply
Date	<b>dt:</b>	You supply
Standard Number	<b>sn:</b>	You supply
Holding Library Code	<b>cd:</b>	You supply
My Status	<b>ms:</b>	You supply
Workflow status	<b>ws:</b>	<b>completed</b> <b>in process</b> <b>in review</b> <b>new</b>

Index	Label	Values
Action Status	<b>as:</b>	delete holdings completed delete holdings failed delete holdings ready delete holdings notset export completed export failed export ready export notset label completed label failed label ready label notset produce completed produce failed produce ready produce notset replace completed replace failed replace ready replace notset update holdings completed update holdings failed update holdings ready update holdings notset validate completed validate failed validate notset
Source Status	<b>ss:</b>	derived extracted imported oclc workform
Format	<b>ft:</b>	books computer files maps mixed materials scores serials sound recordings visual materials
Held Status	<b>hl:</b>	true false

Index	Label	Values
Library	<b>lb:</b>	dlc pcc nlm gpo agl ukm nlc
Retrieval order	<b>ro:</b>	save file number oclc control number date/time added name title

Note: If you record rather than write a macro, statuses are recorded with no spaces between words. When you write macros, you can retain the spaces. Values you enter are not case-sensitive.

### Local bibliographic constant data searches (database identifier **BC**)

See examples for online bibliographic constant data searches (except use the database identifier **BC**).

Indexes and labels:

- Constant data name (**nm:**) (**default**)
- My Status (**ms:**)

### Online bibliographic review records (database identifier **BR**)

Use to find bibliographic records sent to your library for review by another library you specify by institution symbol. (Same as selecting **Review File Records for Institution** check box and entering institution symbol in the Search Online Bibliographic Save File window via **Cataloging > Search > Online Save File**).

Index and label:

Records sent for review (**sm**)

Enter a valid institution symbol (in uppercase) for a library whose records you want to retrieve for review.

**Example:**

```
num = CS.Search("BR", "sm:OCL")
```

```
MsgBox("Found " & num & "records")
```

### Authority File searches (database identifier **AF**)

Guidelines:

- Enter an index label (see list below), a colon, and then the search term.

**Example:**

```
CS.Search "AF", "pn:mitchell"
```

- You can combine searches using Boolean operators. To distinguish the words *and*, *or*, or *not* when they are part of your search term, you must enclose them in single quotation marks (').

**Example:**

**CS.Search "AF", "ti:war 'and' peace"**

- The search is not case-sensitive: you can enter the search string in upper- or lowercase.

Indexes and labels:

Index	Label
Personal Names	<b>pn:</b> (default)
Corporate Names	<b>co:</b>
Conference Names	<b>cn:</b>
Title	<b>ti:</b>
Topicals	<b>to:</b>
Geographic Names	<b>gg:</b>
Children's Subjects	<b>sj:</b>
LCCN	<b>ln:</b>
ISBN	<b>bn:</b>
ISSN	<b>in:</b>
LCSH	<b>su:</b>
OCLC ARN	<b>an:</b>
Subdivision Headings	<b>sb:</b>
LC Name	<b>nw:</b>
Derived Personal Name	<b>pd:</b>
Derived Coporate/Conference/Geographic	<b>cd:</b>
Derived Title	<b>td:</b>
Derived Topical Subject	<b>sd:</b>

**Online authority save file searches** (database identifier AS)

Guidelines:

- Enter an index label (see list below), a colon, and then the search term.
- You can combine searches using Boolean operators. To distinguish the words **and**, **or**, or **not** when they are part of your search term (not used as Boolean operators), you must enclose in single quotation marks.

**Example:**

**CS.Search "AS", "ti:red 'and' the black"**

- The search is not case-sensitive: you can enter the search string in upper- or lowercase.\
- To retrieve all records in the save file, leave the search string parameter empty: **CS.Search "AS", ""**.

Indexes and labels:

Index	Label	Values
Save File Number	<b>sf:</b> (default)	You supply
Personal Names	<b>pn:</b>	You supply
Corporate Names	<b>co:</b>	You supply
Conference Names	<b>cn:</b>	You supply

Index	Label	Values
Titles	<b>ti:</b>	You supply
Topicals	<b>to:</b>	You supply
Geographic Names	<b>gg:</b>	You supply
Children's Subjects	<b>sj:</b>	You supply
LCCN	<b>ln:</b>	You supply
LCSH	<b>su:</b>	You supply
OCLC ARN	<b>an:</b>	You supply
Subdivision Headings	<b>sb:</b>	You supply
Replace Date	<b>rp:</b>	You supply
Workflow Status	<b>ws:</b>	<b>complete</b> <b>in process</b> <b>in review</b> <b>new</b> <b>responded</b> <b>imported</b>
Review Status	<b>rs:</b>	<b>submitted</b>

**Note:** If you record rather than write a macro, statuses are recorded with no spaces between words. When you write macros, you can retain the spaces. Values you enter are not case-sensitive.

#### Online authority constant data searches (database identifier AD)

Search examples:

**CS.Search "AD", "nm:authconstantdata"** (retrieves a constant data record named *authconstantdata*)

**CS.Search "AD", ""** (retrieves all authority constant data records)

Index and label:

Constant Data Name (**nm**) (**default**)

#### Local authority save file (database identifier AL)

Follow the same guidelines as for online authority save file searches (except use the database identifier **AL**).

Index	Label	Values
Save File Number	<b>sf:</b> (default)	You supply
OCLC ARN	<b>an:</b>	You supply
Heading	<b>he:</b>	You supply
Workflow Status	<b>ws:</b>	<b>complete</b> <b>in process</b> <b>in review</b> <b>new</b>
My Status	<b>ms:</b>	You supply
MARC Organization Code	<b>cd:</b>	You supply
Master Record Locked	<b>lk:</b>	<b>true</b> <b>false</b>

Index	Label	Values
Action Status	<b>as:</b>	add completed add failed add ready add notset replace completed replace failed replace ready replace notset submit completed submit failed submit ready submit notset export completed export failed export ready export notset
Retrieval order	<b>ro:</b>	save file number oclc arn date/time added heading

**Note:** If you record rather than write a macro, statuses are recorded with no spaces between words. When you write macros, you can retain the spaces. Values you enter are not case-sensitive.

**Local authority constant data searches** (database identifier AC)

See examples of online authority constant data searches (except use the database identifier **AC**).

Indexes and labels:

- Constant data (**nm:**) (**default**)
- My Status (**ms:**)

**Online authority review records** (database identifier AR)

Use to find authority records sent to your library for review by another library you specify by institution symbol. (Same as selecting **Review File Records for Institution** check box and entering institution symbol in the Search Online Authority Save File window via **Authorities > Search > Online Save File**).

Index and label:

- Records sent for review (**sm:**).  
 Enter a valid institution symbol (in uppercase) for a library whose records you want to retrieve for review.  
 Example:  
 num = CS.Search("AR", "sm:OCL")  
 MsgBox("Found " & num & "records")

## Browse

<b>Syntax</b>	BOOL = CS.Browse (sDatabase, sIndex, sSearchTerm)
<b>Use to</b>	Browse WorldCat or Authority File.  Same as using the following menu commands: <b>Cataloging &gt; Browse &gt; WorldCat</b> <b>Authorities &gt; Browse &gt; Authority File</b>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sDatabase</b>, enter one of the following database identifiers, enclosed in quotation marks (""): <ul style="list-style-type: none"> <li><b>WC</b> (WorldCat)</li> <li><b>AF</b> (Authority File)</li> </ul> <b>Notes:</b> <ul style="list-style-type: none"> <li>If the database parameter is empty or the database identifier is not supported, the browse defaults to WorldCat.</li> <li>If the index parameter is empty or the index label is not supported, the browse defaults to the browse title index for WorldCat or the personal names index for the Authority File.</li> </ul> </li> <li>For <b>sIndex</b> enter a valid index for the online database you want to browse, enclosed in quotation marks. <b>Note:</b> See notes on browsing WorldCat indexes below. See the list of indexes below for browsing the Authority File.</li> <li>For <b>sBrowseTerm</b> enter a browse string (in upper- or lowercase), enclosed in quotation marks.</li> </ul>
<b>Comments</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the browse is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")     If CS.IsOnline = False Then         CS.Logon "", "", ""     End If     If CS.Browse("WC", "ti", "gone with the wind") = True Then         MsgBox "Browse successful"     Else         MsgBox "Browse not successful"     End If End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Browses the WorldCat title for Gone with the Wind.</li> <li>Returns a message confirming that the browse was successful or unsuccessful.</li> </ul>

## Browsing WorldCat

For indexes, labels, and examples, see “Selected browsable indexes” in Cataloging, Search WorldCat, “Browse WorldCat.”

See [Searching WorldCat Indexes](#) for comprehensive descriptions of indexes.

### Browsing the Authority File: Indexes

Index	Label
Personal Names	<b>pn:</b> (default)
Corporate/Conference Names	<b>co:</b>
Title	<b>ti:</b>
Topicals	<b>to:</b>
Geographic Names	<b>gg:</b>
LCSH	<b>su:</b>
Subdivision Headings	<b>sb:</b>
LC Names	<b>nw:</b>
Children’s Subjects	<b>sj:</b>

## Browse Authority Expanded

<b>Syntax</b>	BOOL = CS.BrowseAuthorityExpanded (nRecOffset, sRootIndex, sRootTerm, sExpanded Term)
<b>Use to</b>	<p>Options:</p> <ul style="list-style-type: none"> <li>Browse the Authority File using an expanded scan. That is, specify a heading and subheading for an Authority File browse.</li> </ul> <p>Same as entering a heading and an expanded term in the Browse Authority File window (<b>Authorities &gt; Browse &gt; Authority File</b>) (nRecOffset = 0).</p> <p>Or</p> <ul style="list-style-type: none"> <li>Expand a scan by specifying a subheading for a specific entry in an Authority File browse results list.</li> </ul> <p>Same as entering an expanded term for an entry in a Browse Authority File List of results (nRecOffset = 1 - xxx = number of entry in browse list)</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>nRecOffset</b>, enter the number <b>0</b> for option 1 described above, or enter <b>1</b> - [any browse list record number] for option 2 described above.</li> <li>For <b>sRootIndex</b>, enter the index label for the index you want to browse. Specify for Option 1 only. (See the list of indexes above for browsing the Authority File.)</li> <li>For <b>sRootTerm</b>, enter a browse string for a heading (in upper- or lowercase), enclosed in quotation marks. Specify for option 1 only.</li> <li>For <b>sExpandedTerm</b>, enter a browse string for a subheading (in upper- or lowercase), enclosed in quotation marks.</li> </ul>
<b>Comments</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the expanded browse is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object</pre>

	<pre> Set CS = CreateObject("Connex.Client") If CS.IsOnline = False Then     CS.Logon "", "", "" End If CS.Browse "AF", "sj", "jones" CS.BrowseAuthorityExpanded "4", "", "", "bible" End Sub         </pre>
<p><b>What this examples does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Browses the Authority File children's subjects index for Jones.</li> <li>• For the fourth entry in the browse results list, expands the browse for the subheading Bible.</li> </ul> <p><b>Alternative:</b> Do the heading browse and subheading expanded scan at the same time:  CS.BrowseAuthorityExpanded "0", "sj", "jones", "bible"</p>

## 7. Connexion client macro commands: Edit records

### AddField

<b>Syntax</b>	BOOL = CS.AddField (nInstance, sData)
<b>Use to</b>	Add a variable field in the current record in correct tag order and specify the order of its appearance among other fields with the same tag number.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>nInstance</b>, enter a number to specify the order of the new field among other fields with the same tag. For example, enter <b>2</b> to insert the new field as the second instance of a repeatable field.</li> <li>For <b>sData</b>, enter the data to put in the new field: Type the three-character field number, followed by two one-character indicators (or substitute spaces if no indicators), followed by the field text. Enclose the data string in quotation marks (“”).</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.AddField 2, "245 Second 245 field"      If CS.AddField(3, "245 Third 245 field") = True Then         MsgBox "245 field successfully added"     End If  End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Adds a new repeatable field 245 as the second instance--that is, following the first instance of field 245--with no indicators (two spaces between the tag number 245 and the text). Adds the text <i>Second 245 field</i> to the new field. (No confirmation message.)</li> <li>Adds a third instance of field 245 with no indicators and the text <i>Third 245 field</i>. If the action is successful, returns the confirmation message <i>245 field successfully added</i>.</li> </ul>

## AddFieldLine

<b>Syntax</b>	BOOL = CS.AddFieldLine (nLineNumber, sData)
<b>Use to</b>	Add a variable field to the current record in a specific field position specified by line number, with each field representing one line.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>nLineNumber</b>, enter the field position (line number) where you want the new field inserted.</li> <li>For <b>sData</b>, enter the data you want to add to the field: Type the three-character field number, followed by two one-character indicators (or substitute spaces if no indicators), followed by the field text. Enclose the data string in quotation marks (“”).</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.AddFieldLine 1, "500 Field added at line position 1"      If CS.AddFieldLine(3, "500 Field added at line position 3") = True Then         MsgBox "New field successfully added"     End If      CS.AddFieldLine 999, "500 Field added as last line of record" End Sub </pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Adds a new field 500 as the first line of the record—that is, as the first variable field—with no indicators (two spaces between the tag number 500 and the text). Adds the text <i>Field added at line position 1</i> to the new field. (No confirmation message.)</li> <li>Adds a new field 500 as the third line of the record with no indicators and the text <i>Field added at line position 3</i>. If the field is added, returns the message <i>New field successfully added</i>.</li> <li>Adds a new field 500 at the end of the record with no indicators and the text <i>Field added as last line of record</i>. (No confirmation message.)</li> </ul>

## ControlHeadingsAll

<b>Syntax</b>	BOOL = CS.ControlHeadingsAll
<b>Use to</b>	In the current record, control all controllable headings that are not already controlled.  Same as using the command <b>Edit &gt; Control Headings &gt; All</b> .
<b>Parameters</b>	None
<b>Comment</b>	You must be logged on to use this macro command.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.  <b>Note:</b> A return of FALSE indicates only that the action was unsuccessful. It does not differentiate specifically whether the command itself failed or whether one or more headings could not be controlled.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.AddFieldLine 1, "650 0Ballet dancing " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 2, "650 0Horseback riding " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 3, "650 0Baton twirling " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 4, "650 0Swimming " &amp; Chr(223) &amp; "v" &amp; " Fiction"      CS.ControlHeadings      All CS.CloseRecord False      CS.Search "WC", "#1"      CS.AddFieldLine 1, "650 0Ballet dancing " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 2, "650 0Horseback riding " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 3, "650 0Baton twirling " &amp; Chr(223) &amp; "v" &amp; " Fiction"     CS.AddFieldLine 4, "650 0Swimming " &amp; Chr(223) &amp; "v" &amp; " Fiction"      If CS.ControlHeadingsAll() = True Then         MsgBox "Control Headings All successful"     End If  End Sub </pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Adds four new fields to the record as the first, second, third, and fourth fields (specified by line number), each containing: <ul style="list-style-type: none"> <li>○ No first indicator (blank space)</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Second indicator 0</li> <li>○ Heading text in subfield a (Line 1: <i>Ballet dancing</i>, Line 2: <i>Horseback riding</i>, Line 3: <i>Baton twirling</i>, and Line 4: <i>Swimming</i>)</li> <li>○ A space</li> <li>○ The subfield delimiter character: Chr(223) (for a list of character codes, see “ASCII and hexadecimal codes for diacritics and special characters”)</li> <li>○ Subfield designation v</li> <li>○ A space</li> <li>○ Heading text (In all 4 lines: <i>Fiction</i>)</li> </ul> <ul style="list-style-type: none"> <li>● Controls all headings in the record.</li> <li>● Closes the record without saving it.</li> <li>● Calls up record number 1 from WorldCat again.</li> <li>● Adds the same four new fields with the same indicators and text in the same position.</li> <li>● Controls all headings in the record and confirms that the headings were controlled by returning the message <i>Control Headings All successful</i>.</li> </ul>
--	---

## UncontrolHeading

<b>Syntax</b>	BOOL = CS.UncontrolHeading
<b>Use to</b>	Uncontrol the controlled heading where the cursor is located to make it available for edit.  Same as using the command <b>Edit &gt; Control Headings &gt; Uncontrol</b> .
<b>Parameters</b>	None
<b>Comment</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If ItemType = 0 Then          CS.CursorRow = 8         CS.CursorColumn = 6          If CS.UncontrolHeading() = True Then             MsgBox "Heading uncontrolled"         Else             MsgBox "Uncontrol failed"         End If     End If End Sub </pre>

<b>What this examples does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor in the sixth column of the eighth row.</li> <li>• Uncontrols the heading where the cursor is located (if the heading is controlled) and either confirms that the heading was uncontrolled by returning the message <i>Heading uncontrolled</i> or returns the message <i>Uncontrol failed</i>.</li> </ul>
--------------------------------	--

## UncontrolAll

<b>Syntax</b>	CS.UncontrolAll
<b>Use to</b>	Uncontrol all headings in the current record.  Same as using the command <b>Edit &gt; Control Headings &gt; Uncontrol All</b> .
<b>Parameters</b>	None
<b>Comment</b>	You must be logged on to run this macro.
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.UncontrolAll  End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor in the sixth column of the eighth row.</li> <li>• Uncontrols all controlled headings in record number 1.</li> </ul>

## IsHeadingControlled

<b>Syntax</b>	BOOL = CS.IsHeadingControlled (nLineNumber)
<b>Use to</b>	Determine whether a heading in a specified line of the current record is controlled.
<b>Parameters</b>	For <b>nLineNumber</b> , enter the line number of the heading field for which you want to check the control status. Each field represents one line.
<b>Comment</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the heading is controlled, or FALSE if not.
<b>Example</b>	Sub Main()

	<pre>Dim CS As Object Set CS = CreateObject("Connex.Client")  If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "WC", "#1"  If CS.IsHeadingControlled (16) = True Then  CS.CursorRow = 16 CS.UncontrolHeading End If  End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor in the sixth column of the eighth row.</li> <li>• Uncontrols the heading in line 16 (sixteenth field) if the heading is controlled.</li> </ul>

## Copy

<b>Syntax</b>	CS.Copy (nStartRow, nStartColumn, nEndRow, nEndColumn)
<b>Use to</b>	Copy specified text to the clipboard.  Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Copy</b> .
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nStartRow</b>, enter a number to specify the first row of text you want to copy (start row offset).</li> <li>• For <b>nStartColumn</b>, enter a number to specify the first column of text you want to copy (start column offset).</li> <li>• For <b>nEndRow</b>, enter a number to specify the last row of text you want to copy (end row offset).</li> <li>• For <b>nEndColumn</b>, enter a number to specify the last column of text you want to copy (end column offset).</li> </ul>
<b>Comment</b>	<ul style="list-style-type: none"> <li>• Offsets are 1-based indexes. If any parameter is 0, the text is not copied.</li> <li>• You cannot specify any part of the fixed field unless it is displayed as variable fields.</li> </ul>
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()  Dim CS As Object Set CS = CreateObject("Connex.Client")  If CS.IsOnline = False Then     CS.Logon "", "", "" End If</pre>

	<pre>CS.Search "WC", "#1"  CS.Copy 2, 1, 2, 10 CS.CursorRow = 5 CS.CursorColumn = 1 CS.Paste  End Sub</pre>
<b>What this examples does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Copies the first 10 characters from the second line of the record and pastes at the beginning of the fifth line.</li> </ul>

## Cut

<b>Syntax</b>	BOOL = CS.Cut (nStartRow, nStartColumn, nEndRow, nEndColumn)
<b>Use to</b>	Remove the specified text from its current position and copy to the Windows clipboard.  Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Cut</b> .
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nStartRow</b>, enter a number to specify the first row of text you want to cut (start row offset).</li> <li>• For <b>nStartColumn</b>, enter a number to specify the first column of text you want to cut (start column offset).</li> <li>• For <b>nEndRow</b>, enter a number to specify the last row of text you want to cut (end row offset).</li> <li>• For <b>nEndColumn</b>, enter a number to specify the last column of text you want to cut (end column offset).</li> </ul>
<b>Comment</b>	Offsets are 1-based indexes. If any parameter is 0, the text is not copied. You cannot specify any part of the fixed field unless it is displayed as variable fields.
<b>Return values</b>	None
<b>Example</b>	Works the same as in the example for <b>Copy</b> , except that the 10 characters are removed (not copied) from the second line of the record and pasted at the beginning of the fifth line.

## Paste

<b>Syntax</b>	CS.Paste
<b>Use to</b>	Paste text from the clipboard into the current window at the cursor location.  Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Paste</b> .
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	See the example for <b>Copy</b> , which includes the <b>Paste</b> command.

## CopySelected

<b>Syntax</b>	CS.CopySelected
<b>Use to</b>	Copy selected text to the clipboard
<b>Parameters</b>	None
<b>Comment</b>	You must select text before using this macro command
<b>Return values</b>	None
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If ItemType = 0 Then         CS.FindText "pastimes", 0         CS.CopySelected         CS.CursorRow = 1         CS.CursorColumn = 6         CS.Paste     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Finds and selects the first occurrence of the word <i>pastimes</i> in the record.</li> <li>• Copies <i>pastimes</i>.</li> <li>• Pastes <i>pastimes</i> in the first subfield of the first field in the record.</li> </ul>

## CutSelected

<b>Syntax</b>	CS.CutSelected
<b>Use to</b>	Cut selected text to the clipboard.
<b>Parameters</b>	None
<b>Comment</b>	You must select text before using this macro command.
<b>Return values</b>	None
<b>Example</b>	Works the same as in the example for <b>CopySelected</b> , except the <i>pastimes</i> is removed (not copied) and then pasted.

## CopyField

<b>Syntax</b>	CS.CopyField
<b>Use to</b>	Copy the field where the cursor is located to the clipboard.  Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Copy Field</b> .
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If ItemType = 0 Then         CS.CursorRow = 5         CS.CursorColumn = 1         CS.CopyField     End If End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Locates the cursor at the beginning of the fifth field and copies the entire field.</li> </ul>

## CopyControlNumber

<b>Syntax</b>	CS.CopyControlNumber
<b>Use to</b>	Copy the control number of the current record to the clipboard (control number is the OCLC number of a bibliographic record or the ARN of an authority record).  Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Copy Control Number</b> .
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If</pre>

	<pre>CS.Search "WC", "#1"  CS.CopyControlNumber  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Copies the OCLC control number to the Windowsclipboard.</li> </ul>

## CursorPosition

<b>Syntax</b>	CS.CursorPosition (nPosition)
<b>Use to</b>	Used by the macro recorder to record the position of the cursor when you click to place it in a record.
<b>Parameters</b>	<b>nPosition</b> is a number representing the location of the cursor.
<b>Comments</b>	<p>CursorPosition generally works reliably only for the macro recorder. Its numeric value varies depending on the presence or absence of indicators.</p> <p><b>Recommendation:</b></p> <p>For writing macros, OCLC recommends that you use <b>CursorColumn</b>, instead of CursorPosition to place the cursor. Writing CursorPosition into a macro can cause unexpected results.</p> <p>CursorColumn, on the other hand, reliably places the cursor by the exact number of data points in a field. A data point can be a character or a space; therefore, the presence or absence of indicators is accounted for.</p>
<b>Return values</b>	An integer representing the position of the cursor.

## DeleteCurrentField

<b>Syntax</b>	CS.DeleteCurrentField
<b>Use to</b>	<p>Delete the field where the cursor is located.</p> <p>Same as using the command <b>Edit &gt; Cut Copy Paste Delete Field</b>.</p>
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If</pre>

	<pre> CS.Search "WC", "#1"  If ItemType = 0 Then     CS.CursorRow = 5     CS.CursorColumn = 1     CS.DeleteCurrentField End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• If the result is an open WorldCat record, locates the cursor at the beginning of the fifth field and deletes the entire field.</li> </ul>

## DeleteCurrentWord

<b>Syntax</b>	CS.DeleteCurrentWord
<b>Use to</b>	Delete the word where the cursor is located.
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If ItemType = 0 Then         CS.CursorRow = 17         CS.CursorColumn = 20         CS.DeleteCurrentWord     End If End Sub                 </pre>
<b>What this example does</b>	<p>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</p> <p>Searches WorldCat for record number 1.</p> <p>If the result is an open WorldCat record, places the cursor in the word in the 17th row at the 20th character position and deletes the word.</p>

## DeleteToEndOfCell

<b>Syntax</b>	CS.DeleteToEndOfCell
<b>Use to</b>	Delete text from the cursor position to the end of the cell.
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If ItemType = 0 Then         CS.CursorRow = 3         CS.CursorColumn = 6         CS.DeleteToEndOfCell     End If End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• If the result is an open WorldCat record, places the cursor in the sixth character position in the third field of the record (row 3, column 6) and deletes the text to the end of the cell, starting with the sixth character.</li> </ul>

## MoveFieldUp

<b>Syntax</b>	CS.MoveFieldUp
<b>Use to</b>	<p>Move the field where the cursor is located up one position.</p> <p>Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Move Field &gt; Up</b>.</p>
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"</pre>

	<pre>                 If ItemType = 0 Then                     CS.CursorRow = 3                     CS.CursorColumn = 1                     CS.MoveFieldUp                 End If             End Sub         </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• If the result is an open WorldCat record, places the cursor at the beginning of the third field and moves the field up one position.</li> </ul>

## MoveFieldDown

<b>Syntax</b>	CS.MoveFieldDown
<b>Use to</b>	<p>Move the field where the cursor is located up one position.</p> <p>Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Move Field &gt; Down</b>.</p>
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	Works the same as in the example for MoveFieldUp, except that, instead of moving the field up, moves the field down one position.

## DeleteField

<b>Syntax</b>	BOOL = CS.DeleteField (sTag, nInstance)
<b>Use to</b>	<p>Delete a field by specific tag number and, in the case of repeated variable fields, by order of appearance among the other fields with the same tag number.</p> <p>Same as using the command <b>Edit &gt; Cut Copy Paste &gt; Delete Field</b>.</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sTag</b>, enter the tag of the field you want to delete.</li> <li>• For <b>nInstance</b>, enter a number to specify which instance of a repeated field you want to delete. For example, enter <b>2</b> to delete the second instance of the repeated field.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1" </pre>

	<pre> CS.DeleteField "700", 2  If CS.DeleteField("700", 1) = True Then     MsgBox "700 field successfully deleted" End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Deletes the second instance of field 700. (No confirmation message.)</li> <li>• Deletes the first instance of field 700 and, if successful, returns the confirmation message <i>700 field successfully deleted</i>.</li> </ul>

## DeleteFieldLine

<b>Syntax</b>	BOOL = CS.DeleteFieldLine (nLineNumber)
<b>Use to</b>	Delete a field in a field position specified by line number, with each field representing one line.
<b>Parameters</b>	For <b>nLineNumber</b> , enter the field position (line number) of the field you want to delete.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.DeleteFieldLine 1      If CS.DeleteFieldLine(1) = True Then         MsgBox "The first line was deleted"     End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Deletes the first variable field in the record (line number 1) (no confirmation message)</li> <li>• Deletes the next variable field in the record and, if successful, returns the message <i>The first line was deleted</i>.</li> </ul>

## GetField

<b>Syntax</b>	BOOL = CS.GetField (sTag, nInstance, sData)
<b>Use to</b>	Get the data for a field by tag number and, in the case of repeated variable fields, by order of appearance among the other fields with the same tag number.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sTag</b>, enter the field number of the field you want.</li> </ul> <p><b>Note:</b> You can use a pound sign (#) in a tag number as a wildcard.</p> <p><b>Examples:</b>  <i>CS.GetField("6##", 2, sField)</i> gets the second 6XX tag  <i>CS.GetField("#00", 2, sField)</i> gets the second X00 tag  <i>CS.GetField("###", 2, sData)</i> gets the second line in the record and is the same as using:  <i>CS.GetFieldLine(2, sData)</i></p> <ul style="list-style-type: none"> <li>For <b>nInstance</b>, enter a number to specify which instance of the repeated field you want. For example, enter <b>2</b> to get the second instance of the field.</li> <li>Enter the parameter <b>sData</b> to store the text of the field you specify.</li> </ul>
<b>Comments</b>	<ul style="list-style-type: none"> <li>The first parameter, <b>sTag</b>, accepts the pound sign (#) as a wildcard.</li> <li>Retrieving data that contains non-Latin script characters using this macro command returns a message that "Data contains non-Latin script." Use the <i>GetFieldUnicode</i> command instead for retrieving data with non-Latin scripts.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sData as String      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.GetField "700", 2, sData      MsgBox sData      If CS.GetField("520", 1, sData) = True Then         MsgBox sData     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Gets the second instance of field 700 and stores the data in the field.</li> </ul>

	<ul style="list-style-type: none"> <li>• Returns a message giving the data of field 700.</li> <li>• Gets the first instance of field 520 and returns a message giving the data in the field.</li> </ul>
--	---

## GetFieldLine

<b>Syntax</b>	BOOL = CS.GetFieldLine (nLineNumber, sData)
<b>Use to</b>	Get the data for a field in a field position specified by line number, with each field representing one line.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nLineNumber</b>, enter the field position (line number) of the field you want.</li> <li>• Enter the parameter <b>sData</b> to store the data from the field you specify.</li> </ul>
<b>Comments</b>	Retrieving data that contains non-Latin script characters using this macro command returns a message that "Data contains non-Latin script." Use the GetFieldLineUnicode command instead for retrieving data with non-Latin scripts.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim sData as String      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.GetFieldLine 4, sData      MsgBox sData      If CS.GetFieldLine(10, sData) = True, Then         MsgBox sData     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Gets the data in the fourth variable field of the record (line number 4) and returns a message giving the data.</li> <li>• Gets the data in the tenth variable field of the record (line number 10) and returns a message giving the data.</li> </ul>

## GetFieldUnicode

<b>Syntax</b>	BOOL = CS.GetFieldUnicode (sFieldTag, shFieldInst, sFieldData)
<b>Use to</b>	This command works like GetField, but also returns Unicode data in Numeric Character Reference (NCR) form.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sFieldTag</b>, enter the field number you want to work with.</li> <li>• For <b>nFieldInst</b>, enter a number to specify which instance of a repeatable field you want. For example, enter <b>2</b> to get the second instance of the field.</li> <li>• Enter the parameter <b>sFieldData</b> to store the returned field data.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim nLen As Integer     Dim sFieldData as String      Set CS = CreateObject("Connex.Client")      CS.GetFieldUnicode 245, 1, sFieldData      MsgBox sFieldData      sFieldData = Mid\$(sFieldData, 6, Len(sFieldData))     sFieldData = "500 " + sFieldData      CS.AddFieldLine1, sFieldData End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Gets field 245 in the displayed record as Unicode.</li> <li>• Displays the data in a message box.</li> <li>• Constructs a 500 field from the data.</li> <li>• Adds the new field to the record.</li> </ul>

## GetFieldLineUnicode

<b>Syntax</b>	BOOL = CS.GetFieldLineUnicode (shLineNum, sFieldData)
<b>Use to</b>	This command works like GetFieldLine, but also returns Unicode data in Numeric Character Reference (NCR) form.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>shLineNum</b>, enter the record line number (each field is counted as a line).</li> <li>• Enter the parameter <b>sFieldData</b> to store the returned field data.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim nLen As Integer     Dim sFieldData as String      Set CS = CreateObject("Connex.Client")</pre>

	<pre> CS.GetFieldLineUnicode 8, sFieldData  MsgBox sFieldData  sFieldData) = Mid\$(sFieldData, 6, Len(sFieldData)) sFieldData = "500 " + sFieldData  CS.AddFieldLine1, sFieldData End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Gets field 245 in the displayed record as Unicode.</li> <li>• Displays the data in a message box.</li> <li>• Constructs a 500 field from the data.</li> <li>• Adds the new field to the record.</li> </ul>

## SetField

<b>Syntax</b>	<pre> BOOL = CS.SetField (nInstance, sData) </pre>
<b>Use to</b>	Add a field or add data to an existing field specified by tag number and, in the case of repeated variable fields, by the order of appearance among the other fields with the same tag number.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nInstance</b>, enter a number to specify the position of the new field among the repeated instances of the field. For example, enter <b>2</b> to insert the new field below the first instance of the specified field.</li> <li>• For <b>sData</b>, enter the text you want to insert: Type the three-character field number, followed by two one-character indicators (or substitute spaces if no indicators), followed by the field text. Enclose the data string in quotation marks ("").</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()    Dim CS As Object   Set CS = CreateObject("Connex.Client")    If CS.IsOnline = False Then     CS.Logon "", "", ""   End If    CS.Search "WC", "#1"    CS.SetField 1, "245 New 245 field"    If CS.SetField(2, "700 New second occurrence of 700 field") = True Then     MsgBox "Second 700 field changed"   End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> </ul>

	<ul style="list-style-type: none"> <li>• Adds a new field 245 as the first instance of the field, with no indicators (two spaces between tag number and text) and the text <i>New 245 field</i>.</li> <li>• Adds a new field 700 as the second instance, with no indicators and the text <i>New second occurrence of 700 field</i>. If the action is successful, returns the message <i>Second 700 field changed</i>.</li> </ul>
--	--

## SetFieldLine

<b>Syntax</b>	BOOL = CS.SetFieldLine (nLineNumber, sData)
<b>Use to</b>	Add a field or add data to an existing field in a field position specified by line number, with each field representing one line.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nLineNumber</b>, enter the field position (line number) of the field to which you want to add data.</li> <li>• For <b>sData</b>, enter the data you want to add: Type the three-character field number, followed by two one-character indicators (or use spaces if no indicators), followed by the field text. Enclose the data string in quotation marks (“”).</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.SetFieldLine "1", "500    New first line"      If CS.SetFieldLine(10, "500 New 10th line") = True Then         MsgBox "Line 10 changed"     End If      If CS.SetFieldLine(10, "500 New last line in record") = True Then         MsgBox "New last line added"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Adds a new field 500 as the first field (line) in the record, with no indicators (two spaces between tag number and text) and the text <i>New first line</i>.</li> <li>• Adds a new field 500 as the tenth field in the record, with no indicators and the text <i>New 10th line</i>. If the action is successful, returns the message <i>Line 10 changed</i>.</li> <li>• Adds a new field 500 as the last line of the record, with no indicators and the text <i>New last line in record</i>. If the line is added successfully, returns the message <i>New last line added</i>.</li> </ul>

## SetFixedField

<b>Syntax</b>	BOOL = CS.SetFixedField (sFieldName, sData)
<b>Use to</b>	Insert a value in a specific fixed-field element.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sFieldName</b>, enter the name of the fixed-field element where you want to insert a value, enclosed in quotation marks (“ ”)</li> </ul> <p><b>Note:</b> For <b>Dates 2</b>, enter a comma (,) for the fixed-field element name.  <b>Example:</b> CS.SetFixedField(",", "uuuu")</p> <ul style="list-style-type: none"> <li>For <b>sData</b>, enter the value you want to insert in the specified fixed-field element, enclosed in quotation marks (“ ”)</li> </ul> <p><b>Tip:</b> Enter a space instead of data in the quotation marks to clear the specified fixed-field element..</p>
<b>Comments</b>	<p>This command works only when the setting for displaying the fixed field (<b>View &gt; OCLC Fixed Field</b>) is set to <b>Top</b> or <b>Bottom</b>.</p> <p>The command does not work when the fixed field display is set to <b>Variable Field</b>.</p> <p>See the description of “FixedFieldPosition,” the command for determining or specifying how the fixed field displays.</p>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not. Also returns FALSE if you set a fixed field element as a space to clear it.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.SetFixedField "Type", "t"      If CS.SetFixedField("Lang", "Esp") = True Then         MsgBox "Lang changed to Esp"     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Inserts value <i>t</i> in fixed-field element <i>Type</i>. (No confirmation message.)</li> <li>Inserts value <i>Esp</i> in fixed-field element <i>Lang</i>. If the action is successful, returns the message <i>Lang changed to Esp</i>.</li> </ul>

## GetFixedField

<b>Syntax</b>	BOOL = CS.GetFixedField (sFieldName, sField)
<b>Use to</b>	Get the data for a specific fixed-field element.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sFieldName</b>, enter the name of the fixed-field element you want, enclosed in quotation marks (“”).</li> </ul> <p><b>Note:</b> For Dates 2, enter a comma (,) for the fixed-field element name enclosed in quotation marks.</p> <p><b>Example:</b> CS.GetFixedField “”, sField</p> <ul style="list-style-type: none"> <li>Enter the parameter <b>sField</b> to store the value from the fixed field element you specify.</li> </ul>
<b>Comments</b>	<p>This command works only when the setting for displaying the fixed field (<b>View &gt; OCLC Fixed Field</b>) is set to <b>Top</b> or <b>Bottom</b>.</p> <p>The command does not work when the fixed field display is set to <b>Variable Field</b>.</p> <p>See the description of “FixedFieldPosition,” the command for determining or specifying how the fixed field displays.</p>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sField as String      Set CS = CreateObject(“Connex.Client”)      If CS.IsOnline = False Then         CS.Logon “”, “”, “”     End If      CS.Search “WC”, “#1”      CS.GetFixedField “Type”, sField      MsgBox “Type = “ &amp; sField      If CS.GetFixedField(“Lang”, sField) = True Then         MsgBox “Lang = “ &amp; sField     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Gets the value of fixed-field element Type and returns the message <i>Type = [value]</i>.</li> </ul>

	<ul style="list-style-type: none"> <li>Gets the value of fixed-field element Lang and returns the message <i>Lang = [value]</i>.</li> </ul>
--	---

## FixedFieldPosition

<b>Syntax</b>	CS.FixedFieldPosition
<b>Use to</b>	Return or specify how the fixed field displays in a record.
<b>Parameters</b>	None
<b>Return values</b>	Returns an integer indicating the way the fixed field displays: <ul style="list-style-type: none"> <li><b>0</b> - Fixed field displays as variable fields</li> <li><b>1</b> - Fixed field displays at the top of the record</li> <li><b>2</b> - Fixed field displays at the bottom of the record</li> </ul>
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.FixedFieldPosition = 0 Then         CS.FixedFieldPosition = 1      Select Case CS.FixedFieldPosition()     Case 0         MsgBox "Fixed field displays as variable fields"     Case 1         MsgBox "Fixed field displays at the top of the record"     Case 3         MsgBox "Fixed field displays at the bottom of the record"     End Select End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>If the fixed field displays as variable fields, changes display to fixed-field elements positioned at the top of the record and returns the integer 1.</li> <li>Checks how the client displays the fixed field and returns a message: <ul style="list-style-type: none"> <li>If the fixed field displays as variable fields (Case 0), returns a message <i>Fixed field displays as variable fields</i>.</li> <li>If the fixed-field elements display at the top of the record (Case 1), returns a message <i>Fixed field displays at the top of the record</i>.</li> <li>If the fixed-field elements display at the bottom of the record (Case 2), returns a message <i>Fixed field displays at the bottom of the record</i>.</li> </ul> </li> </ul> <p>In this case, returns the second message above.</p>

## SetCursorFixedField

<b>Syntax</b>	CS.SetCursorFixedField
<b>Use to</b>	Move the cursor to the fixed-field element specified by name and select the value of the fixed field element.
<b>Parameters</b>	<p>For the parameter <b>sFieldName</b>, type the name of the fixed-field element you want to select, enclosed in quotation marks ("").</p> <p><b>Note:</b> Use the name given in <i>Bibliographic Formats and Standards</i>, "Input Standards for Fixed-Field Elements and 006."</p> <p><b>Examples:</b></p> <ul style="list-style-type: none"> <li>• <b>BLvl</b> (Bibliographic Level)</li> <li>• <b>Rec stat</b> (Record Status)</li> <li>• <b>Type</b> (Type of Record)</li> </ul>
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.FixedFieldPosition = 1 Then         CS.SetCursorFixedField "Lang"         If CS.InsertText("ita") = True Then             MsgBox "Value for fixed field Lang added"         End If     End If End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• If the fixed field displays at the top of the record, adds the value ita to the fixed-field element <b>Lang</b>.</li> <li>• Checks how the client displays the fixed field and returns a message:</li> <li>• If the value is added successfully, returns the message <i>Value for fixed field Lang added</i>.</li> </ul>

## GetSelectedText

<b>Syntax</b>	BOOL = CS.GetSelectedText (sText)
<b>Use to</b>	Get selected text from a record (text can span multiple fields).
<b>Parameters</b>	Enter the parameter <b>sText</b> to store the selected text.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.

<b>Comments</b>	You must have a record open and have text selected to run this macro.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim sText as String      If CS.GetSelectedText(sText) = True Then         MsgBox sText     Else         MsgBox "No text is selected"     End If End Sub </pre>
<b>What this example does</b>	Gets selected text from a record and returns a message giving the text, or returns the message <i>No text is selected</i> .

## SetSelectedText

<b>Syntax</b>	BOOL = CS.SetSelectedText (sText)
<b>Use to</b>	Insert specific text in a record at the cursor location (text can span multiple fields).
<b>Parameters</b>	For <b>sText</b> , enter the text string you want to insert, enclosed in quotation marks ("").
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.SetSelectedText "New text" End Sub </pre>
<b>What this example does</b>	Inserts the text <i>New text</i> in a record at the cursor location.

## InsertText

<b>Syntax</b>	BOOL = CS.InsertText (sText)
<b>Use to</b>	Insert specific text in a record at the cursor location (text can span multiple fields).
<b>Parameters</b>	For the parameter <b>sText</b> , enter the text string you want to insert, enclosed in quotation marks ("").
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If End Sub </pre>

	<pre> End If  CS.Search "WC", "#1"  If CS.ItemType = 0 Then     CS.CursorRow = 8     CS.CursorColumn = 6  If CS.UncontrolHeading = True Then     CS.InsertText "Uncontrolled" Else     CS.IndertText "Could not control" End If End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• If the current window is a bibliographic record, places the cursor in the sixth position of the eight row.</li> <li>• If the heading at that position is controlled, uncontrols it and inserts the text <i>Uncontrolled</i>. Otherwise, inserts the text <i>Could not uncontrol</i>.</li> </ul>

## FindText

<b>Syntax</b>	BOOL = CS.FindText (sText, bMatchCase)
<b>Use to</b>	Find and highlight the next occurrence of specific text.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sText</b>, enter the text you want to find, enclosed in quotation marks ("").</li> <li>• For <b>bMatchCase</b>, enter the boolean value to specify whether you want to match upper- and lowercase exactly or not match case: <ul style="list-style-type: none"> <li>○ Match upper- and lowercase exactly</li> <li>○ False - do not match case</li> </ul> </li> </ul>
<b>Comments</b>	<ul style="list-style-type: none"> <li>• FindText should precede ReplaceText.</li> <li>• FindText finds and highlights text only within a single cell of a field, not across cells. You can find tags or indicators or data separately, but not in combination.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 1     CS.CursorColumn = 1                 </pre>

	<pre> If CS.FindText("Rand", False) Then     If CS.GetSelectedText(sText) Then         MsgBox sText     End If Else     MsgBox "Rand - not found" End If  nResult = CS.FindText("rand", True)  If nResult = True Then     If CS.GetSelectedText(sText) Then         MsgBox sText     End If Else     MsgBox "rand - not found" End If  CS.FindText "Favorite", False  If CS.GetSelectedText(sText) Then     MsgBox sText End If End Sub </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Moves the cursor to the first column of the first row (beginning of the record).</li> <li>• Searches the record for the text <i>Rand</i>, disregarding upper- and lowercase (case-insensitive). If found, gets and returns the first occurrence of the text or returns the message <i>Rand - not found</i>.</li> <li>• Searches the record for the text <i>rand</i>, matching case exactly (case-sensitive). If found, gets and returns the first occurrence of the text or returns the message <i>Rand - not found</i>.</li> <li>• Searches the record for the text <i>Favorite</i> (case-insensitive). If found, gets and returns the first occurrence of the text.</li> </ul>

## ReplaceText

<p><b>Syntax</b></p>	<p>BOOL = CS.ReplaceText (sOldText, sNewText, bMatchCase)</p>
<p><b>Use to</b></p>	<p>Replace the first or next occurrence of "found" text with specified text.</p>
<p><b>Parameters</b></p>	<ul style="list-style-type: none"> <li>• For <b>sOldText</b>, enter the text you want to find, enclosed in quotation marks (").</li> <li>• For <b>sNewText</b>, enter the text you want to replace the "found" text, enclosed in quotation marks (").</li> <li>• For <b>bMatchCase</b>, enter the boolean value to specify whether to match upper- and lowercase exactly or not match case:             <ul style="list-style-type: none"> <li>○ True - Match upper- and lowercase exactly</li> <li>○ False - Do not match case</li> </ul> </li> </ul>
<p><b>Comments</b></p>	<ul style="list-style-type: none"> <li>• Precede ReplaceText by FindText.</li> </ul>

	<ul style="list-style-type: none"> <li>• After text is replaced, the cursor remains to the right of the replaced text.</li> <li>• ReplaceText replaces text only within a single cell of a field, not across cells. You can replace tags or indicators or data separately, but not in combination.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim sText as String     Dim nResult As Integer      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 1     CS.CursorColumn = 1      If CS.FindText("XXX", False) = False Then         MsgBox "XXX - not found"     End If      If CS.replaceText("Rand", "XXX", False) Then         MsgBox "The first instance of Rand was replaced with XXX"     End If      nResult = CS.ReplaceText("xxx", "YYY", True)      If nResult - True Then         MsgBox "Incorrect xxx - replaced"     Else         MsgBox "Correct xxx - not replaced"     End If  End Sub         </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor in the first column of the first row (beginning of the record).</li> <li>• Searches the record for the text <i>XXX</i>, disregarding upper- and lowercase (not case- sensitive), and returns the message <i>XXX - not found</i> if the text is notfound.</li> <li>• Searches for the text <i>Rand</i>. If it is found, replaces the first instance of <i>Rand</i> with the text <i>XXX</i>, and returns the message <i>The first instance of Rand was changed to XXX</i>.</li> <li>• Finds and replaces the first instance of the text <i>xxx</i> with the text <i>YYY</i>, matching case exactly. If the action is successful (lowercase <i>xxx</i> is replaced with <i>YYY</i>), returns the message <i>Incorrect xxx - replaced</i>. If the <i>xxx</i> text found does not match case exactly, returns the message <i>Correct xxx - not replaced</i>.</li> </ul>

## ReplaceTextAll

<b>Syntax</b>	CS.ReplaceTextAll (sOldText, sNewText, bMatchCase)
<b>Use to</b>	Replace all occurrences of “found” text with specified text.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sOldText</b>, enter the text you want to find, enclosed in quotation marks (“”).</li> <li>• For <b>sNewText</b>, enter the text you want to replace the “found” text, enclosed in quotation marks (“”).</li> <li>• For <b>bMatchCase</b>, enter the boolean value to specify whether to match upper-and lowercase exactly or not match case: <ul style="list-style-type: none"> <li>○ True - Match upper- and lowercase exactly</li> <li>○ False - Doenot match case</li> </ul> </li> </ul>
<b>Comments</b>	ReplaceTextAll replaces text only within a single cell of a field, not across cells. You can replace all specified tags or indicators or data separately, but not in combination.
<b>Return values</b>	An integer representing the number of replacements.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim sText As String     Dim nNumReplaced As Integer      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 1     CS.CursorColumn = 1      If CS.FindText("XXX", False) = False Then         MsgBox "XXX - not found"     End If      If CS.ReplaceTextAll("Rand", "XXX", False) &gt; 0 Then         MsgBox "All instances of Rand were replaced with XXX"     End If      nNumReplaced = CS.ReplaceTextAll("xxx", "YYY", True)      If nNumReplaced &gt; 0 Then         MsgBox "Incorrect xxx - replaced"     Else         MsgBox "Correct xxx - not replaced"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> </ul>

	<ul style="list-style-type: none"> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor in the first column of the first row (beginning of the record).</li> <li>• Searches the record for the text <i>XXX</i>, disregarding upper- and lowercase (case-insensitive), and returns the message <i>XXX - not found</i> if the text is notfound.</li> <li>• Searches for the text <i>Rand</i>, disregarding upper- and lowercase (case-insensitive). If it is found, replaces all instances of <i>Rand</i> with the text <i>XXX</i>, and returns the message <i>All instances of Rand were replaced with XXX</i>. Also returns an integer representing the number of replacements.</li> <li>• Finds and replaces all instances of the text <i>xxx</i> with the text <i>YYY</i>, matching case exactly. If the action is successful (lowercase <i>xxx</i> is replaced), returns the message <i>Incorrect xxx - replaced</i>. If the <i>xxx</i> text found does not match case exactly, returns the message <i>Correct xxx - not replaced</i>. Also returns an integer representing the number of replacements.</li> </ul>
--	--

## FindReplaceNextText

<b>Syntax</b>	BOOL = CS.FindReplaceNextText (sOldText, sNewText, bMatchCase)
<b>Use to</b>	<p>Find the next occurrence of specified text and select it.</p> <p>Another instance of FindReplaceNextText replaces the selected text, moves to the next occurrence, and selects it.</p> <p>This macro command combines FindText and Replace commands and is used primarily for the macro recorder..</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sOldText</b>, enter the text you want to find, enclosed in quotation marks (“”).</li> <li>• For <b>sNewText</b>, enter the text you want to replace the “found” text, enclosed in quotation marks (“”).</li> <li>• For <b>bMatchCase</b>, enter the boolean value to specify whether to match upper-and lowercase exactly or not match case:             <ul style="list-style-type: none"> <li>○ True - Match upper- and lowercase exactly</li> <li>○ False - Doenot match case</li> </ul> </li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim sText As String     Dim nResult As Integer      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 9     CS.CursorColumn = 1         </pre>

	<pre>If CS.FindReplaceNextText("Rand", "McNally", True) &gt; False Then     MsgBox "Rand not found" End If  If CS.FindReplaceNextText("Rand," "McNally," True) = False Then     MsgBox "Rand not found" End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Moves the cursor to the first column of the ninth row.</li> <li>• Searches the record for the text <i>Rand</i>, matching case exactly (case-sensitive). Selects it if found, or returns the message <i>Rand not found</i>.</li> <li>• Replaces the selected occurrence of <i>Rand</i> with the text <i>McNally</i>, matching case exactly, and then finds and selects the next occurrence of <i>Rand</i>, if found, or returns the message <i>Rand not found</i>.</li> </ul>

## Cancel Changes

<b>Syntax</b>	BOOL = CS.CancelChanges
<b>Use to</b>	<p>Cancel all changes made to a master record (if from WorldCat), or cancels all changes made since the last time you saved a record (if from the save file).</p> <p>Same as using the <b>Edit &gt; Cut Copy Paste &gt; Cancel Changes</b> menu command.</p>
<b>Parameters</b>	None
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.AddField 2, "245 Second 245 field"      CS.CancelChanges  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Adds a second instance of field 245 with no indicators (two spaces between the tag number and field data) and the text <i>Second 245 field</i>.</li> <li>• Cancels the change (deletes the newly added field 245).</li> </ul>

## Reformat

<b>Syntax</b>	BOOL = CS.Reformat
<b>Use to</b>	Reformat a record (record redisplay with fields in correct order).  Same as using the <b>Edit &gt; Reformat</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	You must have a record open to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()  Dim CS As Object Set CS = CreateObject("Connex.Client")  CS.Reformat End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Reformats the current record in correct tag order.</li> <li>• Does not reset the automatic logoff timer.</li> </ul>

## Validate

<b>Syntax</b>	nNumErrors = CS.Validate (sErrorList)
<b>Use to</b>	Validate the current record, or records selected in a list, and return a list of validation errors, if any.  Same as using the <b>Edit &gt; Validate</b> menu command.
<b>Parameters</b>	Enter the parameter <b>sErrorList</b> to store a list of validation errors, if any.
<b>Comments</b>	<ul style="list-style-type: none"> <li>• Errors in the list are separated by a pipe character (   ).</li> <li>• You must be logged on to validate records.</li> <li>• When you validate a record, the client automatically saves it, whether the validation is successful or not.</li> </ul>
<b>Return values</b>	An integer showing the number of validation errors. <b>Note:</b> A return of -1 indicates that validation failed because of a system error.
<b>Example</b>	<pre>Sub Main()  Dim CS As Object Dim sErrorList As String Dim sError As String Dim nNumErrors As Integer Dim nIndex As Integer  Set CS = CreateObject("Connex.Client")  If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "WC", "#1"</pre>

	<pre> ' Create 2 invalid fields CS.AddField 1, "100aa100 field" CS.AddField 1, "500bb500 field"  nNumErrors = CS.Validate(sErrorList)  If nNumErrors &gt; 0 Then     MsgBox "There were " + (nNumErrors) +         " validation errors"  ' Display all of the validation errors MsgBox sErrorList  ' Display the errors individually nIndex = Instr(sErrorList, " ")  While nIndex &gt; 0     MsgBox Left(sErrorList, nIndex + 1)      sErrorList = Mid(sErrorList, nIndex + 1)     nIndex = Instr(sErrorList, " ") Wend  MsgBox sErrorList End If End Sub                 </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Adds two invalid fields: a new first instance of field 100--with invalid indicators <i>aa</i> and the text <i>100 field</i>--and a new first instance of field 500--with invalid indicators <i>bb</i> and the text <i>500 field</i>.</li> <li>• Runs validation on the record and returns the number of errors (two in this case).</li> <li>• Sends the message <i>There were 2 validation errors</i>.</li> <li>• Displays each validation error individually.</li> <li>• Returns a list of errors separated by the pipe ( ) character (in this case two errors about invalid indicators).</li> </ul>

## CursorColumn

<b>Syntax</b>	CS.CursorColumn
<b>Use to</b>	Indicate or change the column where the cursor is located in a record.
<b>Parameters</b>	None
<b>Comments</b>	<p>The leftmost column is column 1 (that is, the first character of the MARC tag). CursorColumn moves the cursor exactly one data point to the right. A data point can be a character or a space. CursorColumn 1, 2, and 3 are always tag characters; 4 and 5 are always indicators, even if blank; and 6 is the start of actual data in the field.</p> <p><b>Note:</b></p>

	<p>CursorColumn works differently than CursorPosition. When writing macros, use CursorColumn to place the cursor by an exact number of data points in a field. CursorPosition is generally useful only for the macro recorder to record the position of the cursor when you manually click to place it in a record. See comments in the description of CursorPosition for more details.</p>
<b>Return values</b>	An integer representing the column number where the cursor is located.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 1     CS.CursorColumn = 4  End Sub         </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Moves the cursor to the fourth column of the first row of the record (the position of the first indicator in the first field).</li> </ul>

## CursorRow

<b>Syntax</b>	CS.CursorRow
<b>Use to</b>	Indicate or change the row where the cursor is located in a record. The top row is row 1. A row corresponds to a field, regardless of how many lines are in a field.
<b>Parameters</b>	None
<b>Return values</b>	An integer representing the row number where the cursor is located.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.CursorRow = 2     CS.CursorColumn = 2  End Sub         </pre>

<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Moves the cursor to the second column of the second row of the record.</li> </ul>
-------------------------------	--

## InsertMode

<b>Syntax</b>	BOOL = CS.InsertMode
<b>Use to</b>	Indicate or change the insert mode (insert mode on or off).
<b>Parameters</b>	None
<b>Comments</b>	The record window must be the current (top-most window) for this command to work.
<b>Return values</b>	TRUE if the insert mode is on, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim bInsertMode As Integer      bInsertMode = CS.InsertMode      If bInsertMode = True Then         MsgBox "Insert mode is on"     Else         MsgBox "Insert mode is off"     End If      CS.InsertMode = False      If CS.InsertMode = True Then         MsgBox "Insert mode is on"     Else         MsgBox "Insert mode is off"     End If End Sub         </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Checks to see if the insert mode is on and returns the message <i>Insert mode is on</i>, or if not, returns the message <i>Insert mode is off</i>.</li> <li>• Turns the insert mode off.</li> <li>• Checks to see if the insert mode is on and returns a message. In this case, the program will return the message <i>Insert mode is off</i>, since the previous command turned it off.</li> </ul>

## ApplyOnlineDefaultCD

<b>Syntax</b>	BOOL = CS.ApplyOnlineDefaultCD
<b>Use to</b>	Apply the default constant data record to the currently displayed bibliographic or authority record.  Same as using the menu command <b>Edit &gt; Constant Data &gt; Online &gt; Apply Default</b> .
<b>Parameters</b>	None
<b>Comments</b>	You must have a list of constant data records open to use this command.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.ApplyOnlineDefaultCD= True Then         MsgBox "Constant data successfully applied"     Else         MsgBox "Constant data not applied"     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Applies the default constant data to record number 1. If successful, returns the message <i>Constant data successfully applied</i>. If not, returns the message <i>Constant data not applied</i>.</li> </ul>

## ApplyOnlineCDByName

<b>Syntax</b>	BOOL = CS.ApplyOnlineCDByName (sName, nOption)
<b>Use to</b>	Apply a constant data record by name to the currently displayed bibliographic or authority record.  Same as using the menu command <b>Edit &gt; Constant Data &gt; Online &gt; Apply by Name</b> .
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sName</b>, enter the name of the constant data record you want to apply.</li> <li>• For <b>nOption</b>, enter one of the following numbers to indicate which fields to apply:             <ul style="list-style-type: none"> <li>○ <b>0</b> - Fixed field</li> <li>○ <b>1</b> - Variable fields</li> <li>○ <b>2</b> - Both</li> </ul> </li> </ul>
<b>Comments</b>	You must have a list of constant data records open to use this command

<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.ApplyOnlineCDBByName("MyCD", 2) = True Then         MsgBox "Constant data successfully applied"     Else         MsgBox "Constant data not applied"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Applies both the fixed field and variable fields of the constant data record named <i>MyCD</i> to WorldCat record number 1. If successful, returns the message <i>Constant data successfully applied</i>. If not, returns the message <i>Constant data not applied</i>.</li> </ul>

## ApplyCDFFromList

**Note:** Before version 1.20 of Connexion client, this macro command was called *ApplyOnlineCDFFromList*. Both versions of this command work. You **do not need to correct macros** you wrote or recorded before you upgraded to version 1.20. This command works for both online and local constant data lists.

<b>Syntax</b>	BOOL = CS.ApplyCDFFromList ( nListIdx, nOption)
<b>Use to</b>	<p>Apply a constant data record from a list to the currently displayed bibliographic or authority record.</p> <p>Same as using one of the following the menu commands:</p> <ul style="list-style-type: none"> <li>• <b>Edit &gt; Constant Data &gt; Online &gt; Apply from List</b></li> <li>• <b>Edit &gt; Constant Data &gt; Local &gt; Apply from List</b>.</li> </ul>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>nListIdx</b>, enter the list number of the constant data record you want to apply (first record on the list is number 1).</li> <li>• For <b>nOption</b>, enter one of the following numbers to indicate which fields to apply: <ul style="list-style-type: none"> <li>○ <b>0</b> - Fixed field</li> <li>○ <b>1</b> - Variable fields</li> <li>○ <b>2</b> - Both</li> </ul> </li> </ul>
<b>Comments</b>	<p>You must have a list of constant data records open to use this command.</p> <p>The list index is 1-based.</p>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.

<p><b>Example</b></p>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.ApplyCDFromList(3, 1) = True Then         MsgBox "Constant data successfully applied"     Else         MsgBox "Constant data not applied"     End If  End Sub         </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Applies variable fields only of the third record in the current list of constant data records to WorldCat record number 1.</li> <li>• If successful, returns the message <i>Constant data successfully applied</i>. If not, returns the message <i>Constant data not applied</i>.</li> </ul>

## ApplyLocalDefaultCD

<p><b>Syntax</b></p>	<p>BOOL = CS.ApplyLocalDefaultCD</p>
<p><b>Use to</b></p>	<p>Apply the local default constant data record to the current record or to records selected in a list.</p> <p>Same as using the menu command <b>Edit &gt; Constant Data &gt; Local &gt; Apply Default</b>.</p>
<p><b>Parameters</b></p>	<p>None</p>
<p><b>Return values</b></p>	<p>TRUE if the action is successful, or FALSE if not.</p>
<p><b>Example</b></p>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.ApplyLocalDefaultCD         </pre>

	<pre>Dim nSlotNum as Integer nSlotNum = CS.SaveToLocalFile (False, False) MsgBox "Record saved with " &amp; nSlotNum &amp; " slot number"</pre> <p>End Sub</p>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Applies the default local constant data record number 1.</li> <li>• Saves the online record number 1 to the default bibliographic local save file.</li> <li>• Displays the following confirmation message with the local save file number (xxx) assigned to the record: <i>Record saved with xxx slot number</i>.</li> </ul>

## ApplyLocalCDBByName

<b>Syntax</b>	<pre>BOOL = CS.ApplyLocalCDBByName (sName, nOption)</pre>
<b>Use to</b>	<p>Apply a local constant data record by name to the current record or to records selected in a list.</p> <p>Same as using the menu command <b>Edit &gt; Constant Data &gt; Local &gt; Apply from List</b>.</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sName</b>, enter the name of the constant data record you want to apply.</li> <li>• For <b>nOption</b>, enter one of the following numbers to indicate which fields to apply: <ul style="list-style-type: none"> <li>○ <b>0</b> - Fixed field</li> <li>○ <b>1</b> - Variable fields</li> <li>○ <b>2</b> - Both</li> </ul> </li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.ApplyCDBByName "MyCD", 2      Dim nSlotNum as Integer     nSlotNum = CS.SaveToLocalFile (False, False)     MsgBox "Record saved with " &amp; nSlotNum &amp; " slot number"</pre> <p>End Sub</p>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> </ul>

	<ul style="list-style-type: none"> <li>• Applies the local constant data record named MyCD to record number 1.</li> <li>• Saves record number 1 to the default bibliographic local save file.</li> <li>• Displays the following confirmation message with the local save file number (xxx) assigned to the record: <i>Record saved with xxx slot number..</i></li> </ul>
--	--

## ChangeRecordType

<b>Syntax</b>	BOOL = CS.ChangeRecordType (nRecordType)
<b>Use to</b>	Change the MARC format of a record to another type.
<b>Parameters</b>	<p>For <b>nRecordType</b>, enter one of the following numbers, depending on which format you want to apply to the current record:</p> <ul style="list-style-type: none"> <li><b>0</b> = Books</li> <li><b>1</b> = Computer Files</li> <li><b>2</b> = Continuing Resources (formerly called Serials; includes serials and integrating resources material types)</li> <li><b>3</b> = Maps</li> <li><b>4</b> = Mixed Materials</li> <li><b>5</b> = Scores</li> <li><b>6</b> = Sound Recordings</li> <li><b>7</b> = Visual Materials</li> </ul>
<b>Return values</b>	TRUE if the record changes to the specified format, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.ChangeRecordType 1 = True Then         MsgBox "Record format changed to computer files"     Else         MsgBox "Constant data not applied"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Changes record type to Computer Files.</li> <li>• If the action is successful, returns the message <i>Record format changed to computer files</i>.</li> </ul>

## InsertFromCitedRecord

<b>Syntax</b>	BOOL = CS.InsertFromCitedRecord
<b>Use to</b>	<p>Insert data from a cited record that has a bibliographic connection to the record you are editing.</p> <p>Same as using the <b>Edit &gt; Insert from Cited Record</b> menu command. See details in the section "Build fields that cite related records (760 - 787)" in the Edit Bibliographic Records guide.</p>
<b>Parameters</b>	None
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      bReturn = CS.InsertFromCitedRecord     If bReturn = true Then         MsgBox "Insert From Cited Record succeeded"     Else         MsgBox "Insert From Cited Record failed !!!"     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Inserts data from the record identified by OCLC control number in the field (760 - 787, 800, 810, 811, or 830) that you created.</li> <li>• If the action is successful, returns the message <i>Insert From Cited Record succeeded</i>. If not, returns the message <i>Insert From Cited Record failed !!!</i></li> </ul>

## GetClassNumber

<b>Syntax</b>	BOOL = CS.GetClassNumber (classScheme, classNum)
<b>Use to</b>	<p>Retrieve a classification number for the displayed record. Uses the prototype OCLC Classify Web service (at <a href="http://classify.oclc.org/">http://classify.oclc.org/</a>) to retrieve a call number by matching records based on the following standard numbers, in this order: OCLC number, ISBN, ISSN, UPC, LCCN.</p> <p>Similar to using the <b>Edit &gt; Classify</b> menu command. See more details about how this function works in "Get a classification number for a bibliographic record" in Cataloging, Edit Bibliographic Records guide.</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• Use <b>classScheme</b> to specify the Dewey Decimal Classification scheme—enter zero (0)—or the Library of Congress classification scheme—enter one (1).</li> <li>• <b>classNum</b> contains the classification number returned from the Classify service if a single class number is retrieved.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim classNum As String</pre>

	<pre> Dim CS As Object Set CS = CreateObject("Connex.Client")      If CS.GetClassNumber(1, classNum) = True Then         sField = "090 " &amp; classNum     If CS.AddField (99, sField) = True Then         CS.Reformat     End If Else     MsgBox "GetClassNumber returned False" End If End Sub         </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Sends standard numbers (OCLC control number, ISBN, ISSN, UPC, LCCN) in the displayed bibliographic record to the Classify Web service for matching (in the order listed) and extracts a Dewey classification number.</li> <li>• If the Classify service finds a single match, the command:</li> <li>• Returns <i>True</i> with the class number in the classNum parameter</li> <li>• Creates an 090 field with the classification number inserted (using "AddField")</li> <li>• Adds the new field to the end of the record</li> <li>• Reformats the record in MARC tag order</li> <li>• If the Classify service does not find a single match:</li> <li>• Returns <i>False</i> with a blank in the classNum parameter</li> <li>• Returns the message <i>GetClassNumber returned False</i></li> </ul>

## 8. Connexion client macro commands: Create records

### DeriveNewRecord

<b>Syntax</b>	BOOL = CS.DeriveNewRecord (bFixedField)
<b>Use to</b>	Derive a new record from the current record.  Same as using the menu command <b>Edit &gt; Derive &gt; New Master Record</b> .
<b>Parameters</b>	For <b>bFixedField</b> , enter <b>True</b> if you want to transfer the fixed field to the new record, or enter <b>False</b> if not. <b>Notes:</b> <ul style="list-style-type: none"> <li>The command uses your settings in <b>Tools &gt; Options &gt; Derive Record</b> to determine the variable fields to transfer to the new record (<b>default:</b> 1XX through 8XX, where <b>X</b> is any number).</li> <li>Although this parameter does not apply to authority records, you must enter a value for the macro to derive an authority record successfully. Enter either <b>True</b> or <b>False</b>. The client ignores the parameter for derived authority records.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim bResult As Integer      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.DeriveNewRecord(True) = True Then         MsgBox "Derive was successful"     Else         MsgBox "Derive failed"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Derives a new record based on record number 1, transfers the fixed field to the new record, and if the action is successful, returns the message <i>Derive was successful</i>. If not, returns the message <i>Derive failed</i>.</li> </ul>

## OpenWorkform

<b>Syntax</b>	BOOL = CS.OpenWorkform (sFormat)																																		
<b>Use to</b>	<p>Open a bibliographic workform to create a record of a specified MARC format.</p> <p>Same as using the menu commands:</p> <ul style="list-style-type: none"> <li>• Cataloging &gt; Create &gt; Single Record &gt; Books</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Computer Files</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Continuing Resources</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Maps</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Mixed Materials</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Scores</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Sound Recordings</li> <li>• Cataloging &gt; Create &gt; Single Record &gt; Visual Materials</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Blank Record</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Conference Name</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Corporate Name</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Geographics</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Personal Name</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Series</li> <li>• Authorities &gt; Create &gt; Single Record &gt; Uniform Title</li> </ul>																																		
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sFormat</b>, enter the type of format you want the workform to have, enclosed in quotation marks. Enter one of the following values for bibliographic workforms:</li> </ul> <table border="0" style="margin-left: 20px;"> <tr> <td><b>Enter</b></td> <td><b>For this format type</b></td> </tr> <tr> <td>bks</td> <td>Books</td> </tr> <tr> <td>com</td> <td>Computer Files</td> </tr> <tr> <td>cnr*</td> <td>Continuing Resources</td> </tr> <tr> <td>map</td> <td>Maps</td> </tr> <tr> <td>mix</td> <td>Mixed Materials</td> </tr> <tr> <td>sco</td> <td>Scores</td> </tr> <tr> <td>rec</td> <td>Sound Recordings</td> </tr> <tr> <td>vis</td> <td>Visual Materials</td> </tr> </table> <p>*Formerly <b>ser</b> (Serials); you can use either <b>cnr</b> or <b>ser</b>.</p> <ul style="list-style-type: none"> <li>• Enter one of the following values for authority workforms:</li> </ul> <table border="0" style="margin-left: 20px;"> <tr> <td><b>Enter</b></td> <td><b>For this format type</b></td> </tr> <tr> <td>abl</td> <td>Blank Workform</td> </tr> <tr> <td>acn</td> <td>Conference Name</td> </tr> <tr> <td>acr</td> <td>Corporate Name</td> </tr> <tr> <td>age</td> <td>Geographics</td> </tr> <tr> <td>apn</td> <td>Personal Name</td> </tr> <tr> <td>asr</td> <td>Series</td> </tr> <tr> <td>aut</td> <td>Uniform Title</td> </tr> </table>	<b>Enter</b>	<b>For this format type</b>	bks	Books	com	Computer Files	cnr*	Continuing Resources	map	Maps	mix	Mixed Materials	sco	Scores	rec	Sound Recordings	vis	Visual Materials	<b>Enter</b>	<b>For this format type</b>	abl	Blank Workform	acn	Conference Name	acr	Corporate Name	age	Geographics	apn	Personal Name	asr	Series	aut	Uniform Title
<b>Enter</b>	<b>For this format type</b>																																		
bks	Books																																		
com	Computer Files																																		
cnr*	Continuing Resources																																		
map	Maps																																		
mix	Mixed Materials																																		
sco	Scores																																		
rec	Sound Recordings																																		
vis	Visual Materials																																		
<b>Enter</b>	<b>For this format type</b>																																		
abl	Blank Workform																																		
acn	Conference Name																																		
acr	Corporate Name																																		
age	Geographics																																		
apn	Personal Name																																		
asr	Series																																		
aut	Uniform Title																																		
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.																																		

## OpenWorkFormForDatabase

<b>Syntax</b>	BOOL = CS.OpenWorkformForDatabase (sFormat) (sDatabase)
<b>Use to</b>	Open a workform to create a record of a specified MARC type for a specified database, including WorldCat, and LC authority file.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For sFormat, enter the type of format you want the workform to have, enclosed in quotation marks. Enter one of the following values for a WorldCat workform: <ul style="list-style-type: none"> <li>○ <b>bks</b> for Books</li> <li>○ <b>com</b> for Computer Files</li> <li>○ <b>cnr*</b> for Continuing Resources</li> <li>○ <b>map</b> for Maps</li> <li>○ <b>mix</b> for Mixed Materials</li> <li>○ <b>sco</b> for Scores</li> <li>○ <b>rec</b> for Sound Recordings</li> <li>○ <b>vis</b> for Visual Materials</li> </ul> <p>*Formerly <b>ser</b> (Serials). You can use either <b>cnr</b> or <b>ser</b>.</p> <ul style="list-style-type: none"> <li>○ Enter one of the following values for authority workforms: <ul style="list-style-type: none"> <li>○ <b>abl</b> for a blank workform</li> <li>○ <b>acn</b> for Conference Name</li> <li>○ <b>acr</b> for Corporate Name</li> <li>○ <b>age</b> for Geographics</li> <li>○ <b>apn</b> for Personal Name</li> <li>○ <b>asr</b> for Series</li> <li>○ <b>aut</b> for Uniform Title</li> </ul> </li> </ul> </li> <li>• For sDatabase, enter <b>WC</b> for WorldCat, <b>AF</b> for the LC authority file.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.

## 9. Connexion client macro commands: Take actions on records

### UpdateHoldings

<b>Syntax</b>	BOOL = CS.UpdateHoldings
<b>Use to</b>	For an existing record or a workform, or records or workforms selected in a list, update holdings (add your OCLC institution symbol). For workform(s), add to WorldCat. Or When offline, marks local record(s) for batch update. Same as using the <b>Action &gt; Holdings &gt; Update Holdings</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	<ul style="list-style-type: none"> <li>If you are logged on, this macro does the action immediately online.</li> <li>If you are working with local file records while offline, this macro marks the current record, or records selected in a list, ready (<b>R</b>) to batch-process the action.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      bResult = CS.UpdateHoldings      If bResult = True Then         MsgBox "Holdings were successfully updated"     Else         MsgBox "Holdings were not successfully updated"     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>For record number 1, updates holdings (adds your OCLC institution symbol).</li> <li>If the action is successful, returns the message <i>Holdings were successfully updated</i>. If not, returns the message <i>Holdings were not successfully updated</i>.</li> </ul>

## ProduceAndUpdateHoldings

<b>Syntax</b>	BOOL = CS.ProduceAndUpdateHoldings
<b>Use to</b>	<p>For an existing record or a workform, or for records or workforms selected in a list, update holdings (adds your OCLC institution symbol to the records) and produce offline catalog cards as specified in your OCLC cataloging profile. If for workform(s), add to WorldCat.</p> <p>Or</p> <p>When offline, marks local record(s) for batch update and produce.</p> <p>Same as using the <b>Action &gt; Holdings &gt; Produce and Update Holdings</b> menu command.</p>
<b>Parameters</b>	None
<b>Comments</b>	<ul style="list-style-type: none"> <li>• If you are logged on, this macro does the action immediately online.</li> <li>• If you are working with local file records while offline, this macro marks the current record, or records selected in a list, ready (<b>R</b>) to batch-process the action.</li> </ul>
<b>Return values</b>	TRUE if the actions are successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      bResult = CS.ProduceAndUpdateHoldings      If bResult = True Then         MsgBox "Holdings were successfully updated"     Else         MsgBox "Holdings were not successfully updated"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Updates holdings (adds your OCLC institution symbol) in record number 1.</li> <li>• If the action is successful, returns the message <i>Holdings were successfully updated</i>. If not, returns the message <i>Holdings were not successfully updated</i>.</li> </ul>

## AltProduceAndUpdateHoldings

<b>Syntax</b>	BOOL = CS.AltProduceAndUpdateHoldings (bAdProduce, bAllProduce, shExtraCards)
<b>Use to</b>	<p>For an existing record or workform, or for records or workforms selected in a list, update holdings (add your OCLC institution symbol) and specify AdProduce, AllProduce, and/or the number of extra cards to produce. If workform(s), add to WorldCat.</p> <p>Or</p> <p>When offline, marks local record(s) for batch update and alternate produce.</p> <p>Same as using the <b>Action &gt; Holdings &gt; Alternate Produce and Update Holdings</b> menu command.</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>bAdProduce</b>, enter <b>true</b> to produce standard cards as defined in your OCLC cataloging profile, or enter <b>false</b> to cancel this option.</li> <li>• For <b>bAllProduce</b>, enter <b>true</b> to produce standard cards for all holding libraries listed in the 049 field of the record, or enter <b>false</b> to cancel this option.</li> <li>• For <b>shExtraCards</b>, enter the number of extra copies you want of the main entry card for the record.</li> </ul>
<b>Comments</b>	<ul style="list-style-type: none"> <li>• If you are logged on, this macro does the action immediately online.</li> <li>• If you are working with local file records while offline, this macro marks the current record, or records selected in a list, ready (<b>R</b>) to batch-process the action.</li> </ul>
<b>Return values</b>	TRUE if the actions are successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer     Dim bAdProduce As Integer     Dim bAllProduce As Integer     Dim shExtraCards As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.AltProduceAndUpdateHoldings True, True, 6      CS.Search "WC", "#2"      bAdProduce = True     bAllProduce = False     shExtraCards = 4      bResult = CS.AltProduceAndUpdateHoldings(bAdProduce, bAllProduce,     shExtraCards </pre>

	<pre> If bResult = True Then     MsgBox "The command completed successfully" Else     MsgBox "The command did not complete successfully" End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• For record number 1: <ul style="list-style-type: none"> <li>○ Updates holdings (adds your institution symbol)</li> <li>○ Produces standard cards as defined in your OCLC cataloging profile</li> <li>○ Produces standard cards for all holding libraries listed in the 049 field of the record</li> <li>○ Produces six extra copies of the main entry card</li> </ul> </li> <li>• Searches WorldCat for record number 2.</li> <li>• For record number 2: <ul style="list-style-type: none"> <li>○ Updates holdings (adds your institution symbol)</li> <li>○ Produces standard cards as defined in your OCLC cataloging profile</li> <li>○ Produces four extra copies of the main entry card</li> <li>○ If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul> </li> </ul>

## DeleteHoldings

<b>Syntax</b>	BOOL = CS.DeleteHoldings
<b>Use to</b>	Delete your holdings (OCLC symbol) from the current record or from records selected in a list.  Same as using the <b>Action &gt; Holdings &gt; Delete Holdings</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	<ul style="list-style-type: none"> <li>• If you are logged on, this macro does the action immediately online.</li> <li>• If you are working with local file records while offline, this macro marks the current record, or records selected in a list, ready (<b>R</b>) to batch-process the action.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1" </pre>

	<pre> bResult = CS.DeleteHoldings  If bResult = True Then     MsgBox "The command completed successfully" Else     MsgBox "The command did not complete successfully" End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Deletes holdings (removes your OCLC institution symbol) from record number 1.</li> <li>• If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>

## Lock

<b>Syntax</b>	BOOL = CS.Lock
<b>Use to</b>	<p>Lock a record.</p> <p>Same as using the <b>Action &gt; Lock Master Record</b> menu command.</p>
<b>Parameters</b>	None
<b>Comments</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      bResult = CS.Lock      If bResult = True Then         MsgBox "The command completed successfully"     Else         MsgBox "The command did not complete successfully"     End If End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> </ul>

	<ul style="list-style-type: none"> <li>• Searches WorldCat for record number 1.</li> <li>• Locks the master copy of record number 1.</li> <li>• If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>
--	--

## ReleaseRecordLock

<b>Syntax</b>	BOOL = CS.ReleaseRecordLock
<b>Use to</b>	Release the lock on a master record.  Same as using the <b>Action &gt; Release Record Lock</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	You must be logged on and have a locked record displayed to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.Lock = True Then          bResult = CS.ReleaseRecordLock      If bResult = True Then         MsgBox "The command completed successfully"     Else         MsgBox "The command did not complete successfully"     End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Locks the master copy of record number 1.</li> <li>• Releases the lock on the record.</li> <li>• If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>

## ReplaceRecord

<b>Syntax</b>	BOOL = CS.ReplaceRecord
<b>Use to</b>	<p>Replace an online bibliographic or authority (NACO-authorized libraries) record, or records selected in a list, with the current record.</p> <p>Or</p> <p>When offline, marks local record(s) for batch replace.</p> <p>Same as using the <b>Action &gt; Replace Record</b> menu command.</p>
<b>Parameters</b>	None
<b>Comments</b>	<ul style="list-style-type: none"> <li>• If you are logged on, this macro does the action immediately online.</li> <li>• If you are working with local file records while offline, this macro marks the current record, or records selected in a list, ready (<b>R</b>) to batch-process the action.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.Lock = True Then          CS.AddFieldLine 1, "500 Field added at line position 1"          bResult = CS.ReplaceRecord          If bResult = True Then             MsgBox "The command completed successfully"         Else             MsgBox "The command did not complete successfully"         End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Locks the master copy of record number 1.</li> <li>• Adds a new field 500 as the first line of the record, with no indicators (two blank spaces), and the text Field added at line position 1.</li> <li>• Replaces the master copy of the record (master record now contains the new field).</li> </ul>

	<ul style="list-style-type: none"> <li>If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>
--	---

## ReplaceAndUpdateHoldings

<b>Syntax</b>	BOOL = CS.ReplaceAndUpdateHoldings
<b>Use to</b>	<p>Replace a master bibliographic record, or records selected in a list, with the current record and update holdings (adds your OCLC institution symbol to the records). Or When offline, marks local record(s) for batch update and replace.</p> <p>Same as using the <b>Action &gt; Replace and Update Holdings</b> menu command.</p>
<b>Parameters</b>	None
<b>Comments</b>	<ul style="list-style-type: none"> <li>You must be logged on.</li> <li>You cannot use this macro command to mark local file records for batch processing. This command involves two actions. Batch processing can complete only one action per record.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.Lock = True Then          CS.AddFieldLine 1, "500 Field added at line position 1"          bResult = CS.ReplaceAndUpdateHoldings          If bResult = True Then             MsgBox "The command completed successfully"         Else             MsgBox "The command did not complete successfully"         End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Locks the master copy of record number 1.</li> </ul>

	<ul style="list-style-type: none"> <li>• Adds a new field 500 as the first line of the record, with no indicators (two blank spaces), and the text <i>Field added at line position 1</i>.</li> <li>• Replaces the master copy of the record (master record now contains the new field), and updates holdings (adds your OCLC institution symbol to the record).</li> <li>• If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>
--	---

## SaveOnline

<b>Syntax</b>	nSaveFileNumber = CS.SaveOnline
<b>Use to</b>	Save a record, or records selected in a list, in the online save file.  Same as using the <b>Action &gt; Save Record to Online File</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	You must be logged on to run this macro.
<b>Return values</b>	Save file number(s) the system assigns to the saved record(s).
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim nSaveFileNumber As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nSaveFileNumber = CS.SaveOnline     MsgBox "The record was added as save file number" + (nSaveFileNumber) End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Saves record number 1 to the online bibliographic save file.</li> <li>• Returns the message <i>The record was added as save file number [system-supplied number]</i>.</li> </ul>

## SaveToLocalFile

<b>Syntax</b>	nSaveFileNumber = CS.SaveToLocalFile (bAuthUnlock, bDelete)
<b>Use to</b>	Save a record, or records selected in a list, to the default local save file  Same as using the <b>Action &gt; Save Record to Local File</b> menu command.

<p><b>Parameters</b></p>	<ul style="list-style-type: none"> <li>For <b>bAuthUnlock</b>, enter <b>True</b> to unlock an Authority File record before saving to the local file, or enter <b>False</b> to save the authority record with its lock. <b>Note:</b> You cannot save locked bibliographic records to the local file.</li> <li>For <b>bDelete</b>, enter <b>True</b> to delete a record you save to the local file from the online file, or enter <b>False</b> to keep the record in the online file when it is saved to the local file. <b>Note:</b> Entering values for these parameters is required.</li> </ul>
<p><b>Return values</b></p>	<p>Save file number(s) the system assigns to the saved record(s). Or The value -1 when the corresponding command (<b>Action &gt; Save Record to Local File</b>) is unavailable. The command is unavailable when a record is re-saved to the local file with no changes to the record.</p> <p><b>Tip:</b> If the macro returns -1, you can get the save file number by using:</p> <pre>bool = QueryRecordStatus ("SAVE", sNumber)</pre>
<p><b>Example</b></p>	<pre>Sub Main()      Dim CS As Object     Dim nSaveFileNumber As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search "BS", "" &gt; 0 Then         nNumRecords = CS.SearchList("Mitchell", "Contributor/Creator", False)     End If      nSaveFileNumber = CS.SaveToLocalFile (False, True)      MsgBox "The record(s) were added as save file number(s)" + (nSaveFileNumber) End Sub</pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches the online bibliographic save file for all records.</li> <li>Searches the results list for <i>Mitchell</i> in the <i>Contributor/Creator</i> column, returning terms with different case or that contain the term.</li> <li>Selects the records that match (for the sake of this example, assume that four records are selected).</li> <li>Saves the selected records to the local bibliographic save file and deletes them from the online save file.</li> <li>Returns the message <i>The record(s) were added as local save file numbers 14 15 16 17</i>.</li> </ul>

## DeleteRecord

<b>Syntax</b>	BOOL = CS.DeleteRecord
<b>Use to</b>	Delete a record, or records selected in a list, from the online or local save file.  Same as using the <b>Action &gt; Delete Record</b> menu command.
<b>Parameters</b>	None
<b>Comments</b>	You must be logged on and have a save file record open to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "BS", "sf:1"      bResult = CS.DeleteRecord      If bResult = True Then         MsgBox "The command completed successfully"     Else         MsgBox "The command did not complete successfully"     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches online bibliographic save file for the record with save file number 1.</li> <li>• Deletes save file record number 1.</li> <li>• If the action is successful, returns the message <i>The command completed successfully</i>. If not, returns the message <i>The command did not complete successfully</i>.</li> </ul>

## Export

<b>Syntax</b>	BOOL = CS.Export
<b>Use to</b>	Export the current record, or records selected in a list, using your default export destination from <b>Tools &gt; Options &gt; Export</b> . You can export bibliographic or authority records. Or Marks record(s) for batch export if option is set in <b>Tools &gt; Options &gt; Batch</b> .  Same as using the <b>Action &gt; Export</b> menu command.

<b>Parameters</b>	None
<b>Comments</b>	Whether you are logged on or offline, records export immediately, unless you set the <b>Perform local actions in batch</b> option for <b>Record Export</b> in <b>Tools &gt; Options &gt; Batch</b> to mark records for batch export.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      bResult = CS.Export      If bResult = True Then         MsgBox "Export completed successfully"     Else         MsgBox "Export did not complete successfully"     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Exports the record to your default export destination (<b>Tools &gt; Options &gt; Export</b>).</li> <li>• If the action is successful, returns the message <i>Export completed successfully</i>. If not, returns the message <i>Export did not complete successfully</i>.</li> </ul>

## ExportAppend

<b>Syntax</b>	BOOL = CS.ExportAppend (sName)
<b>Use to</b>	<p>Export record(s) to a destination file specified by its name as pre-defined in <b>Tools &gt; Options &gt; Export</b>.</p> <ul style="list-style-type: none"> <li>• Creates the file if it doesn't already exist and adds the records, or appends the records to the end of an existing file.</li> <li>• Eliminates export prompts.</li> <li>• Exports the displayed record or records selected in a list.</li> </ul> <p><b>Required:</b> The export destination file must be pre-defined in <b>Tools &gt; Options &gt; Export</b> or the macro will fail. Defining the file destination does not create the file. The file is created the first time you export records to it.</p>
<b>Parameters</b>	For <b>sName</b> , enter the name of the export destination file that you assigned when you created the destination. Enclose the name in quotation marks (").

<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.ExportAppend "Data" End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Exports the currently displayed record, or records selected in a list, to the file destination named <i>Data</i>. Creates the file if it doesn't exist.</li> <li>If the file exists and already contains records, appends the newly exported record(s) to the end of the file.</li> <li>No prompts are displayed.</li> </ul>

## AddToAuthorityFile

<b>Syntax</b>	BOOL = CS.AddToAuthorityFile
<b>Use to</b>	<p>Add an authority record, or records selected in an online or local authority list, to the Authority File</p> <p>Or</p> <p>When offline, marks local record(s) for batch add.</p> <p>Same as using the <b>Action &gt; Add to Authority File</b> menu command.</p>
<b>Parameters</b>	None
<b>Return values</b>	<ul style="list-style-type: none"> <li>You must be logged on to run this macro immediately. When offline, marks local save file record(s) ready for batch processing (<b>R</b>).</li> <li>You must be an authorized NACO participant to use this command.</li> </ul>
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "AS", "sf:12"      bResult = CS.AddToAuthorityFile      If bResult = True Then         MsgBox "Record successfully added to the Authority File"     Else         MsgBox "Record not added to the Authority File"     End If End Sub</pre>

<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the online authority save file for save file record number 12, which is an authority workform.</li> <li>• Adds the record to the Authority File.</li> <li>• If successful, returns the message <i>Record successfully added to the Authority File</i>. If not, returns the message <i>Record not added to the Authority File</i>.</li> </ul>
-------------------------------	--

## AddToOnlineDatabase

<b>Syntax</b>	BOOL = CS.AddToOnlineDatabase (sDatabase)
<b>Use to</b>	Add an authority record, or records selected in an online or local authority list, to the LC authority file  Same as using the <b>Action &gt; Add to Authority File</b> menu command.
<b>Parameters</b>	For <b>sDatabase</b> , enter this database identifier: <b>AF</b> (LC names and subjects authority file) (an alternative to using AddToAuthorityFile)
<b>Comments</b>	<ul style="list-style-type: none"> <li>• You must be logged on to run this macro immediately. When offline, marks local save file record(s) ready for batch processing (<b>R</b>).</li> <li>• You must be an authorized NACO participant or an authorized CERL member to use this command for the LC authority file</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.

## SubmitForReview

<b>Syntax</b>	BOOL = CS.SubmitForReview (sSymbol)
<b>Use to</b>	Submit a bibliographic or authority record, or records selected in a list, for peer review to libraries that you specify by one or more OCLC institution symbols. Or When offline, marks local record(s) for batch submit.  Same as using the <b>Action &gt; Submit for Review</b> menu command.
<b>Parameters</b>	For <b>sSymbol</b> , enter one or more OCLC institution symbols. If more than one, separate by semicolons (;). Enter in upper- or lowercase. Spaces are optional. <b>Examples:</b> <b>ABC;DEF;XYZ</b> <b>abc; def; xyz</b>
<b>Comments</b>	<ul style="list-style-type: none"> <li>• You must be logged on to run this macro immediately. When offline, marks local save file record(s) ready for batch processing (<b>R</b>).</li> <li>• Maximum length for the string of institution codes is 255 characters.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")</pre>

	<pre> If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "BS", "sf:1"  bResult = CS.SubmitForReview(ocl;trn)  If bResult = True Then     MsgBox "Record successfully sent for review" Else     MsgBox "Record not sent for review" End If End Sub                 </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the online bibliographic save file for save file record number 1.</li> <li>• Submits the record to two libraries with OCLC symbols <i>ocl</i> and <i>trn</i>.</li> <li>• If successful, returns the message <i>Record successfully sent for review</i>. If not, returns the message <i>Record not sent for review</i>.</li> </ul> <p><b>Note:</b> To locate records that others have sent to your library for review, use <code>CS.Search("BR", "")</code>. See Search macro descriptions for more information.</p>

## RecallSubmitted

<p><b>Syntax</b></p>	<p>BOOL = CS.RecallSubmitted (sSymbol)</p>
<p><b>Use to</b></p>	<p>Recall a record from review.</p> <p>Same as using the <b>Action &gt; Recall Submitted Record</b> menu command.</p>
<p><b>Parameters</b></p>	<p>None</p>
<p><b>Comments</b></p>	<p>You must be logged on to run this macro</p>
<p><b>Return values</b></p>	<p>TRUE if the action is successful, or FALSE if not.</p>
<p><b>Example</b></p>	<pre> Sub Main()      Dim CS As Object     Dim bResult As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search("BS", "rs:submitted") &gt; 0 Then         CS.GetFirstItem     End If      bResult = CS.RecallSubmitted                 </pre>

	<pre> If bResult = True Then     MsgBox "Submitted record successfully recalled from review" Else     MsgBox "Submitted record not recalled from review" End If End Sub                 </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the online bibliographic save file for records your library submitted for review. <b>Note:</b> You can also search for all sum</li> <li>• If the search retrieves multiple records, selects and opens the first record in the list.</li> <li>• Removes the review status "Submitted" from the record and withdraws it from review.</li> <li>• If successful, returns the message <i>Submitted record successfully recalled from review</i>. If not, returns the message <i>Submitted record not recalled from review</i>. <b>Note:</b> You can also search for all submitted records</li> </ul>

## 10. Connexion client macro commands: Work with records, lists, and files

### SearchList

<b>Syntax</b>	nNumRecords = CS.SearchList (sSearchTerm, sColumnName, bExactMatch)
<b>Use to</b>	Search for a specified term in the current list data. You can specify the column name to search. You can also specify to match the search text exactly.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sSearchTerm</b>, enter the search term string</li> <li>For <b>sColumnName</b>, enter the column name string</li> <li>For <b>bExactMatch</b>, enter <b>true</b> to match the search term exactly, or enter <b>false</b> to also return terms with different case or that contain the term</li> </ul> <p><b>Examples:</b> If you specify exact match, the search term <i>Jones</i> does not retrieve <i>jones</i>. If you specify no exact match, the search term <i>free</i> can also retrieve <i>Free</i> or <i>freedom</i>.</p>
<b>Comments</b>	The search is not case-sensitive.
<b>Return values</b>	An integer, the number of records found.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim nNumRecords As Integer      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search("WC", "gon,wi,th,w/bks/1983-2003") &gt; 0 Then         nNumRecords = CS.SearchList("Margaret", "Description", False)          MsgBox (nNumRecords) + " records found"     End If End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for the title <i>Gone with the Wind</i> in Books format only and only items published between 1983 and 2003.</li> <li>If the search retrieves multiple records (a brief list in this case), searches the description column of the list for any instance (not case-sensitive) of the term <i>Margaret</i>.</li> <li>Returns the number of records containing <i>Margaret</i> or <i>margaret</i> in the message <i>X records found</i> (where <i>X</i> = the number of records).</li> <li>Selects the records that contain the search term.</li> </ul>

## GetListItem

<b>Syntax</b>	BOOL = CS.GetListItem (nListIdx)
<b>Use to</b>	Open a record from the currently displayed list by specifying its order in the list by number.
<b>Parameters</b>	For <b>nListIdx</b> , enter the number that represents the order of the record that you want to open in the list.
<b>Comments</b>	The first record on the list is 1 (1-based index).
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search("WC", "gon,wi,th,w") &gt; 0 Then         CS.GetListItem 2     End If End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for the title <i>Gone with the Wind</i>.</li> <li>• If the search retrieves multiple records, opens the second item on the list (in this case, opens the second brief list on a group list).</li> </ul>

## GetFirstSelectedItem

<b>Syntax</b>	BOOL = CS.GetFirstSelectedItem
<b>Use to</b>	Opens the first selected entry on the current list.
<b>Parameters</b>	None
<b>Comments</b>	Opens any type of entry on the list, whether it is a record or another list.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.Logon "", "", ""     CS.Search "WC", "gon,wi,th,w/sco"     CS.SearchList "Max", "Description", False     boolean = CS.GetFirstSelectedItem     While boolean = True         CS.Validate sErrors         CS.CloseRecord False         boolean = CS.GetNextSelectedItem     End While End Sub</pre>

	Wend End Sub
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for items with the title <i>Gone with the Wind</i> in scoresformat.</li> <li>• Searches the <i>Description</i> column of the results list and selects items in the list that contain the term <i>Max</i>. Selects seven items.</li> <li>• Gets the first selected record in the list, validates the record, and closes it without saving.</li> <li>• Gets the next selected record in the list.</li> </ul>

## GetNextSelectedItem

<b>Syntax</b>	BOOL = CS.GetNextSelectedItem
<b>Use to</b>	Select and open the first entry on the current list.
<b>Parameters</b>	None
<b>Comments</b>	Opens any type of entry on the list, whether it is a record or another list.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	See example above for GetFirstSelectedItem.

## GetFirstItem

<b>Syntax</b>	BOOL = CS.GetFirstItem
<b>Use to</b>	Select and open the first entry on the current list.
<b>Parameters</b>	None
<b>Comments</b>	Opens any type of entry on the list, whether it is a record or another list.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      ' Get the first record from the first list     If CS.Search("WC", "gon,wi,th,w") &gt; 0 Then         If CS.GetFirstItem() = True Then             CS.GetFirstItem()         End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for the title <i>Gone with the Wind</i>.</li> </ul>

	<ul style="list-style-type: none"> <li>• If the search retrieves multiple records (a group list in this case), selects and opens the first item on the list (which is a brief list).</li> <li>• Selects and opens the first record in the brief list.</li> </ul>
--	--

## GetNextItem

<b>Syntax</b>	BOOL = CS.GetNextItem
<b>Use to</b>	Select and open the next entry on the current list.
<b>Parameters</b>	None
<b>Comments</b>	<p>Opens any type of entry on the list, whether it is a record or another list.</p> <p>If you are on a record, opens the next record in the list.</p>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      ' Get the first record from the first list     If CS.Search("WC", "gon,wi,th,w") &gt; 0 Then         ' Get brief list         If CS.GetFirstItem() = True Then             ' Get the first record from the list             If CS.GetFirstItem() = True Then                 ' Move to the second record                 CS.GetNextItem() = True Then             End If         End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for the title <i>Gone with the Wind</i>.</li> <li>• If the search retrieves multiple records (a group list in this case), selects and opens the first item on the list (which is a brief list).</li> <li>• Selects and opens the first record in the brief list.</li> <li>• Selects and opens the next record in the list.</li> </ul>

## GetPrevItem

<b>Syntax</b>	BOOL = CS.GetPrevItem
<b>Use to</b>	Select and open the previous item on the current list.
<b>Parameters</b>	None

<b>Comments</b>	Opens any type of entry on the list, whether it is a record or another list.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search("WC", "gon,wi,th,w") &gt; 0 Then         If CS.GetFirstItem() = True Then             If CS.GetFirstItem() = True Then                 If CS.GetNextItem() = True Then                     CS.GetPrevItem                 End If             End If         End If     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for the title <i>Gone with the Wind</i>.</li> <li>• If the search retrieves multiple records (a group list in this case), selects and opens the first item on the list (which is a brief list).</li> <li>• Selects and opens the first record in the brief list.</li> <li>• Selects and opens the next record in the list.</li> <li>• Selects and opens the previous record in the list (the first record opened originally).</li> </ul>

## GetNext100Records

<b>Syntax</b>	BOOL = CS.GetNext100Records
<b>Use to</b>	<p>Open the next list of 100 records resulting from a search of:</p> <ul style="list-style-type: none"> <li>• Online bibliographic save file</li> <li>• Authority File</li> <li>• Online constant data file</li> </ul> <p><b>Note:</b> Search results lists from these online files display 100 records at a time. Same as using the <b>View &gt; Navigate Records and Lists &gt; Next 100 Records</b> menu command.</p>
<b>Parameters</b>	None
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client") </pre>

	<pre> If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "AF", "pn:William"  CS.GetNext100Records End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the Authority File for the heading <i>William</i>. (The search retrieves more than 100 records.)</li> <li>• Gets the next (second set of) 100 records.</li> </ul>

## GetPrevious100Records

<b>Syntax</b>	<pre> BOOL = CS.GetPrevious100Records                 </pre>
<b>Use to</b>	<p>Open the previous list of 100 records resulting from a search of:</p> <ul style="list-style-type: none"> <li>• Online bibliographic save file</li> <li>• Authority File</li> <li>• Online constant data file</li> </ul> <p><b>Note:</b> Search results lists from these online files display 100 records at a time. Same as using the <b>View &gt; Navigate Records and Lists &gt; Previous 100 Records</b> menu command.</p>
<b>Parameters</b>	None
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "AF", "pn:William"      CS.GetNext100Records      CS.GetPrevious100Records End Sub                 </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the Authority File for the heading <i>William</i>. (The search retrieves more than 100 records.)</li> <li>• Gets the next (second set of) 100 records.</li> <li>• Gets the previous 100 records (goes back to the first set).</li> </ul>

## CloseList

<b>Syntax</b>	BOOL = CS.CloseList
<b>Use to</b>	Close the current list.
<b>Parameters</b>	None
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      If CS.Search("WC", "gon,wi,th,w/bks/1983-2003") &gt; 0 Then         If CS.GetFirstItem() = True Then             CS.CloseList         End If     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for the title <i>Gone with the Wind</i> in Books format only and only items published between 1983 and 2003.</li> <li>• If the search retrieves multiple records (a brief list in this case), selects and opens the first item on the list (a record).</li> <li>• Closes the list.</li> </ul>

## CloseRecord

<b>Syntax</b>	CS.CloseRecord (bSaveRecord)
<b>Use to</b>	Close a record.
<b>Parameters</b>	For <b>bSaveRecord</b> , enter <b>true</b> to save the record or <b>false</b> to close without saving.
<b>Comments</b>	You must enter the bSaveRecord parameter (no default value is provided). If you do not, you get an error message.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"</pre>

	<pre>CS.CloseRecord True End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Closes record number 1 and saves it to the online bibliographic save file.</li> </ul>

## GetWindowTitle

<b>Syntax</b>	<pre>BOOL = CS.GetWindowTitle (nWindowId, sTitle)</pre>
<b>Use to</b>	Return the specified window title.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• <b>nWindowId</b> is a zero-based window identification.</li> <li>• Enter the parameter <b>sTitle</b> to store the window title.</li> </ul>
<b>Comments</b>	If you use <b>-1</b> as the window identification, the macro returns the title of the current window.
<b>Return values</b>	TRUE if the title is retrieved, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sTitle As String      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.OpenWorkform "bks"      CS.Search "WC", "#1"     CS.Search "WC", "gon,wi,th,w"      CS.GetWindowTitle 0, sTitle      MsgBox sTitle      CS.SetTopWindow 0      CS.GetWindowTitle -1, sTitle      MsgBox sTitle End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Opens a workform with the Books format.</li> <li>• Searches WorldCat for record number 1 (opens a record).</li> <li>• Searches WorldCat for <i>Gone with the Wind</i> (opens a list).</li> </ul>

	<ul style="list-style-type: none"> <li>• Returns the title of the currently active window: in this case a list, WorldCat Group List, returned from the derived title search for <i>Gone with the Wind</i>.</li> <li>• Changes the top window to the first window opened.</li> <li>• Returns the title of the new top window, the first window opened: in this case, <i>Bibliographic Workform: Books</i>.</li> </ul>
--	--

## ItemType

<b>Syntax</b>	nType = CS.ItemType
<b>Use to</b>	Return the window type of the current (top-most) window.
<b>Parameters</b>	None
<b>Return values</b>	<p>Returns an integer that indicates the type of the current window. The numbers indicate the following window type:</p> <ul style="list-style-type: none"> <li>0 Online bibliographic record (WorldCat)</li> <li>1 Online bibliographic save file record</li> <li>2 Online bibliographic constant data record</li> <li>3 Online authority record (Authority File)</li> <li>4 Online authority save file record</li> <li>5 WorldCat group list</li> <li>6 WorldCat brief list</li> <li>7 Online bibliographic save file list</li> <li>8 Online bibliographic constant data list</li> <li>9 WorldCat browse list</li> <li>10 Authority File brief list</li> <li>11 Authority browse list</li> <li>12 News list</li> <li>13 Online authority save file list</li> <li>14 Online authority constant data record</li> <li>15 Online authority constant data list</li> <li>16 Authority expanded browse list</li> <li>17 Local bibliographic save file record</li> <li>18 Local authoritysave file record</li> <li>19 Local bibliographic constant data record</li> <li>20 Local authority constant data record</li> <li>21 Local bibliographic save file list</li> <li>22 Local authority save file list</li> <li>23 Local bibliographic constant data list</li> <li>24 Local authority constant data list</li> <li>25 WorldCat truncated list</li> <li>27 Online bibliographic institution list</li> <li>28 Unidentified item</li> </ul>
<b>Example</b>	<pre>Sub Main()      Dim CS as Object     Set CS = CreateObject("Connex.Client")      If CS.Logon "", "" "" = True Then         CS.Search("AS", "pn:jones")     End If End Sub</pre>

	<pre> If CS.ItemType = 13 Then   MsgBox "Online authority save file list"   If CS.GetFirstItem() = True Then     If CS.ItemType = 4 Then       MsgBox "Online authority save file record" End If     End If   End If End If End Sub </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches the online authority save file for the personal name <i>Jones</i>.</li> <li>• If the search successfully retrieves multiple records, returns the message <i>Online authority save file list</i>.</li> <li>• Gets the first item on the list and, if the first item is an authoirty save file record, returns the message <i>Online authority save file record</i>.</li> </ul>

## SetTopWindow

<p><b>Syntax</b></p>	<p>BOOL = CS.SetTopWindow (nWindowId)</p>
<p><b>Use to</b></p>	<p>Make the window specified by its numeric identification the current (topmost) window.</p>
<p><b>Parameters</b></p>	<p>For <b>nWindowId</b>, enter the window number you want to make the current window, with 0 (zero) representing the first window.</p>
<p><b>Return values</b></p>	<p>TRUE if the action is successful, or FALSE if not.</p>
<p><b>Example</b></p>	<pre> Sub Main()    Dim CS As Object   Dim sTitle As String    Set CS = CreateObject("Connex.Client")    If CS.IsOnline = False Then     CS.Logon "", "", ""   End If    CS.OpenWorkform "bks"    CS.Search "WC", "#1"    CS.Search "WC", "gon,wi,th,w"    CS.GetWindowTitle -1, sTitle    MsgBox sTitle    CS.SetTopWindow 0    CS.GetWindowTitle -1, sTitle </pre>

	<pre>MsgBox sTitle End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Opens a workform with the Books format.</li> <li>• Searches WorldCat for record number 1 (opens a record).</li> <li>• Searches WorldCat for <i>Gone with the Wind</i> (opens a group list).</li> <li>• Returns the title of the currently active window: in this case <i>WorldCat Group List</i>, returned from the derived title search for <i>Gone with the Wind</i>.</li> <li>• Changes the top window to the first window opened (in this case a workform with the Books format).</li> <li>• Returns the title of the new top window, <i>Bibliographic Workform: Books</i>.</li> </ul>

## WindowCount

<b>Syntax</b>	nCount = CS.WindowCount
<b>Use to</b>	Give the number of open windows.
<b>Parameters</b>	None
<b>Comments</b>	Counts only record and list windows only.
<b>Return values</b>	An integer representing the number of open windows.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.OpenWorkform "bks"      CS.Search "WC", "#1"      CS.Search "WC", "gon,wi,th,w"      MsgBox CStr(CS.WindowCount) + " open windows" End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Opens a workform with the Books format.</li> <li>• Searches WorldCat for record number 1 (opens a record).</li> <li>• Searches WorldCat for <i>Gone with the Wind</i> (opens a group list).</li> <li>• Returns the total number of open record and list windows (in this case 3) in the message <i>3 open windows</i>.</li> </ul>

## CreateLocalFile

<b>Syntax</b>	nResult = CS.CreateLocalFile (sPath)
<b>Use to</b>	Create a local file in the specified path and set the file as the default. If the file already exists, set the specified existing file as the default.
<b>Parameters</b>	For sPath, enter the full path and file name.
<b>Comments</b>	The macro command does not validate the path name for a file, nor does it create directories in the path. The specified file extension determines the type of file created: <ul style="list-style-type: none"> <li>• <b>*.bib.db</b> creates a local bibliographic save file</li> <li>• <b>*.auth.db</b> creates a local authority save file</li> <li>• <b>*.bibcd.db</b> creates a local bibliographic constant data file</li> <li>• <b>*.authcd.db</b> creates a local authority constant data file</li> </ul>
<b>Return values</b>	Returns an integer with one of the following values: <b>0</b> = New file successfully created and set as the default <b>1</b> = Existing file successfully set as the default <b>2</b> = Access to the specified file denied <b>3</b> = Specified file not found <b>4</b> = Undefined error
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      Dim nResult as Integer     nResult = CS.CreateLocalFile "D:\Program Files\MyLocalFiles\ Jan10.bib.db"      If nResult = 0 Then         MsgBox "File was created and set as the default successfully"     Else If nResult = 1 Then         MsgBox "Local file already exists. Set as default"      Else         MsgBox "Error creating local file"     End If End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Attempts to create a new local bibliographic save file named <i>Jan10.bib.db</i> in the file path <i>D:\Program Files\MyLocalFiles</i>.</li> <li>• If the new file is successfully created and set as the default, returns the message <i>File was created and set as the default successfully</i>. Or If the file already exists, sets the file as default and returns the message <i>Local file already exists. Set as default</i>. Or If the file is not created or does not exist or another type of error occurs, returns the message <i>Error creating local file</i>.</li> </ul>

## CopyRecord

<b>Syntax</b>	CS.CopyRecord (sToFile)
<b>Use to</b>	Copy a local file record or records selected in a local file list to another, existing local file that you specify.  Same as using the <b>Action &gt; Copy Record</b> menu command.
<b>Parameters</b>	For <b>sToFile</b> , enter the full path and file name of an existing local file to which you want to copy the record(s).
<b>Comments</b>	The macro command must specify an existing path and file name, or the macro will fail. It does not create folders in the path or the file.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.CopyRecord "d:\Data\NewBib.bib.db" End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Copies the current local record or records selected in a local file list to the <b>NewBib.bib.db</b> file in the \Data folder of the D drive.</li> <li>• Also retains the record(s) in the original file.</li> </ul>

## MoveRecord

<b>Syntax</b>	CS.MoveRecord (sToFile)
<b>Use to</b>	Move a local file record or records selected in a local file list to another, existing local file that you specify.  Same as using the <b>Action &gt; Move Record</b> menu command.
<b>Parameters</b>	For <b>sToFile</b> , enter the full path and file name of an existing local file to which you want to move the record(s).
<b>Comments</b>	The macro command must specify an existing path and file name, or the macro will fail. It does not create folders in the path or the file.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.MoveRecord "d:\Data\NewBib.bib.db" End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Moves the current local record or records selected in a local file list to the <b>NewBib.bib.db</b> file in the \Data folder of the D drive.</li> <li>• Removes the the record(s) from the originalfile.</li> </ul>

## Macro commands for navigating in records

<p><b>Syntax ... use to</b></p>	<p><b>Syntax:</b>                  CS.EndCell                  CS.HomeCell                  CS.EndRecord                  CS.HomeRecord                  CS.KeyUp                  CS.KeyDown                  CS.KeyRight                  CS.KeyLeft                  CS.NextSubfield                  CS.PrevSubfield                  CS.DeleteKey                   CS.Backspace</p> <p><b>Use to move the cursor:</b>                  To the end of the current cell                  To the beginning of the current cell                  To the end of the variable fields                  To the beginning of the variable fields                  One line up                  One line down                  One position to the right                  One position to the left                  To the next subfield                  To the previous subfield                  One position to the right and delete the character or space to the right of the cursor's original position                  One position to the left and delete the character or space to the left of the cursor's original position</p>
<p><b>Parameters</b></p>	<p>None</p>
<p><b>Comments</b></p>	<p>Most of these commands correspond to keys on the keyboard. They are used for themacro recorder.</p>
<p><b>Return values</b></p>	<p>None</p>
<p><b>Example</b></p>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.ItemType = 0 Then         CS.HomeRecord         CS.EndCell         CS.KeyRight         CS.KeyRight         CS.KeyRight         CS.EndCell     End Sub                 </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Places the cursor at the end of the first row of record number 1.</li> </ul>

## 11. Connexion client macro commands: Print and work with macros, statuses, holdings, labels, user profiles

### SetMyStatus

<b>Syntax</b>	BOOL = CS.SetMyStatus (sStatusName)
<b>Use to</b>	Add a customized My Status to the current record.  Same as using the menu command <b>Action &gt; Set Status</b> .
<b>Parameters</b>	For <b>sStatusName</b> , enter the My Status text you want to add to the record. Enclose in quotation marks. Use free text limited to 40 characters (enter in lowercase, uppercase or mixed case).
<b>Comments</b>	You must be logged on to run this macro.  If you enter more than 40 characters, the data is truncated at 40 character.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.OpenWorkform "bks"      CS.SetMyStatus "My workform" End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Opens a workform with a Books format.</li> <li>• Assigns the My Status <i>My workform</i> to the workform.</li> </ul>

### SetWorkflowStatus

<b>Syntax</b>	BOOL = CS.SetWorkflowStatus (sStatusName)
<b>Use to</b>	Change workflow status for the current record (default: <i>In Progress</i> ).  Same as using the menu command <b>Action &gt; Set Status</b> .
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sStatusName</b>, enter one of the following, enclosed in quotation marks (enter in lowercase, uppercase or mixed cast):             <ul style="list-style-type: none"> <li>○ Completed (for bibliographic records) Complete (for authority records)</li> <li>○ In Process</li> <li>○ In Review</li> <li>○ New</li> </ul> </li> </ul>
<b>Comments</b>	You must be logged on to run this macro.

<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.OpenWorkform "bks"      CS.SetWorkflowStatus "New"  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Opens a workform with a Books format.</li> <li>• Assigns the workflow status <i>New</i> to the workform.</li> </ul>

## IsHeldBy

<b>Syntax</b>	BOOL = CS.IsHeldBy (sSymbol, sState)
<b>Use to</b>	Determine whether a specified holding symbol (OCLC institution symbol) is included in the current record.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sSymbol</b>, enter the holding symbol.</li> <li>• Enter the parameter <b>sState</b> to store the state/country of the institution represented by the holding symbol.</li> </ul>
<b>Comments</b>	You must be logged on to run this macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      If CS.IsHeldBy("OCL", sState) = True Then         MsgBox "Record #1 is held by OCL in " + sState     End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> </ul>

	<ul style="list-style-type: none"> <li>Determines whether the institution symbol <i>OCL</i> (symbol for OCLC) is included in the current record and returns a message that includes the state/country where OCLC is located: <i>Record #1 is held by OCL in Ohio [state where OCLC is located].</i></li> </ul>
--	--

## Holdings

<b>Syntax</b>	CS.Holdings
<b>Use to</b>	Return a list of all holdings for the current record.  Same as using the menu command <b>View &gt; Holdings &gt; All</b> .
<b>Parameters</b>	None
<b>Return values</b>	A list of all holding symbols for the current record, separated by commas.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sHoldings As String      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      sHoldings = CS.Holdings      MsgBox Left(sHoldings, 100) End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Returns a list of holding symbols, separated by commas, for record number 1. If the list is more than 100 characters, returns only the first 100 characters.</li> </ul>

## Print

<b>Syntax</b>	BOOL = CS.Print
<b>Use to</b>	Print the current (topmost) record.  Same as using the menu command <b>File &gt; Print</b> .
<b>Parameters</b>	None
<b>Comments</b>	Uses your default printer in Windows, unless you previously set the option in <b>Tools &gt; Options &gt; Printing</b> to print records to a text file..
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	Sub Main()

	<pre> Dim CS As Object Set CS = CreateObject("Connex.Client")  If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "WC", "#1"  If CS.Print() = True Then     MsgBox "Record printed OK" End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Prints record number 1. If the record prints successfully, returns a message <i>Record printed OK</i>.</li> </ul>

## PrintList

<b>Syntax</b>	BOOL = CS.PrintList
<b>Use to</b>	Print the current (topmost) list.  Same as using the menu command <b>File &gt; Print List</b> .
<b>Parameters</b>	None
<b>Comments</b>	Uses your default printer in Windows.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre> Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "gon,wi,th,w"      If CS.PrintList() = True Then         MsgBox "List printed OK"     End If  End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for <i>Gone with the Wind</i> using a derived title search.</li> </ul>

	<ul style="list-style-type: none"> <li>Prints the search results list. If the list prints successfully, returns a message <i>List printed OK</i>.</li> </ul>
--	--

## PrintLabel

<b>Syntax</b>	<p>BOOL = CS.PrintLabel (sHoldingCode, nCopies, sConstant1, sNumbers1, sConstant2, sNumbers2)</p>
<b>Use to</b>	<p>Print the label from the current record or from records selected in a list, based on field data in the record(s). Or If you have the <b>Label Printing</b> option selected under <b>Perform local actions in batch</b> (in <b>Tools &gt; Options &gt; Batch</b>):</p> <ul style="list-style-type: none"> <li>Marks the record ready (<b>R</b>) for batch label printing (if record is in the local save file). Or</li> <li>Saves the record to the local save file and then marks it ready for batch (if record is in the online save file).</li> </ul> <p>Same as using the menu command <b>File &gt; Print Label(s)</b>.</p>
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sHoldingCode</b>, enter the holding library code used to print the label, enclosed in quotation marks (determines the input stamps for the label).</li> <li>For <b>nCopies</b>, enter the number of copies of the label to print (1 to 999). <b>Default:</b> 1 copy.</li> <li>For <b>sConstant1</b>, enter text to print on the label to identify multiple copies or parts, enclosed in quotation marks. Use letters, numbers, and punctuation marks.</li> <li>For <b>sNumbers1</b>, enter numbers to appear on the label with print constant 1, enclosed in quotation marks. Use only numbers, commas, hyphens, and quotation marks.</li> <li>For <b>sConstant2</b>, enter additional text to print on the label, enclosed in quotation marks. Use letters, numbers, and punctuation marks.</li> <li>For <b>sNumbers2</b>, enter numbers to appear on the label with print constant 2, enclosed in quotation marks. Use only numbers, commas, hyphens, and quotation marks.</li> </ul>
<b>Comments</b>	<ul style="list-style-type: none"> <li>You must start the label printing process with <b>StartLabelBatch</b> and end the process with <b>EndLabelBatch</b> when you use the <b>PrintLabel</b> command.</li> <li>Field data in the record(s) that impacts on content of the label(s) must be correct before you use this command.</li> <li>Printing setup options in <b>Tools &gt; Options &gt; Printing</b> apply.</li> <li>Uses your default printer in Windows or the label printer you select in <b>Tools &gt; Options &gt; Printing</b>.</li> </ul> <p>See Cataloging/"Print labels" for more information about how the client produces labels.</p> <ul style="list-style-type: none"> <li>The client automatically saves the record when you use this macro command.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")</pre>

	<pre> If CS.IsOnline = False Then     CS.Logon "", "", "" End If  CS.Search "WC", "gon,wi,th,w"  batchStarted = CS.StartLabelBatch If batchStarted = True Then     CS.PrintLabel "ZMYA", 2, "v.", "1,2,3", "c.", "1,2"     CS.EndLabelBatch End If  End Sub         </pre>
<p><b>What this example does</b></p>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for <i>Gone with the Wind</i> using a derived title search.</li> <li>• Starts a label batch print and prints two copies of labels based on field data for record number 1 for holding library code ZMYA with the following print constants:             <ul style="list-style-type: none"> <li>○ v.1 c.1</li> <li>○ v.1 c.2</li> <li>○ v.2 c.1</li> <li>○ v.2 c.2</li> <li>○ v.3 c.1</li> <li>○ v.3 c.2</li> </ul> </li> </ul>

## PrintLabelEx

<p><b>Syntax</b></p>	<p>BOOL = CS.PrintLabel Ex (sHoldingCode, sSpine, sCall1, sAuthor1, sTitle1, sCall2, sAuthor2, sTitle2, nCopies, sConstant1, sNumbers1, sConstant2, sNumbers2)</p>
<p><b>Use to</b></p>	<p>Allows you to edit data for a label for the current record before printing the label.</p> <p>Same as using the menu command <b>View &gt; Label</b> to open a label for the current record for editing.</p>
<p><b>Parameters</b></p>	<ul style="list-style-type: none"> <li>• For <b>sHoldingCode</b>, enter the holding library code used to print the label, enclosed in quotation marks (determines the input stamps for the label).</li> <li>• Enter the parameter <b>sSpine</b> to supply the text to print on the spine label.</li> <li>• Enter the parameter <b>sCall1</b> to supply the call number to print on the pocket label.</li> <li>• Enter the parameter <b>sAuthor1</b> to supply the author to print on the pocket label.</li> <li>• Enter the parameter <b>sTitle1</b> to supply the title to print on the pocket label.</li> <li>• Enter the parameter <b>sCall2</b> to supply the call number to print on the second pocket label.</li> <li>• Enter the parameter <b>sAuthor2</b> to supply the author to print on the second pocket label.</li> <li>• Enter the parameter <b>sTitle2</b> to supply the title to print on the second pocket label.</li> <li>• For <b>nCopies</b>, enter the number of copies of the label to print (1 to 999). <b>Default:</b> 1 copy.</li> <li>• For <b>sConstant1</b>, enter text to print on the label to identify multiple copies or parts, enclosed in quotation marks. Use letters, numbers, and punctuation marks.</li> </ul>

	<ul style="list-style-type: none"> <li>• For <b>sNumbers1</b>, enter numbers to appear on the label with print constant 1, enclosed in quotation marks. Use only numbers, commas, hyphens, and quotation marks.</li> <li>• For <b>sConstant2</b>, enter additional text to print on the label, enclosed in quotation marks. Use letters, numbers, and punctuation marks.</li> <li>• For <b>sNumbers2</b>, enter numbers to appear on the label with print constant 2, enclosed in quotation marks. Use only numbers, commas, hyphens, and quotation marks.</li> </ul>
<p><b>Comments</b></p>	<ul style="list-style-type: none"> <li>• You must have a record open to use this macro command.</li> <li>• Use this command with GetLabelSpine, GetLabelCallNumber, GetLabelAuthor, and GetLabelTitle to retrieve and edit data for the content of a label before printing.</li> <li>• Leave pocket 2 parameters empty to duplicate pocket 1 parameters (same as selecting</li> <li>• <b>Mirror Edits in View &gt; Label</b>).</li> <li>• You must start the label printing process with <b>StartLabelBatch</b> and end the process with <b>EndLabelBatch</b> when you use the <b>PrintLabelEx</b> command.</li> <li>• Printing setup options in <b>Tools &gt; Options &gt; Printing</b> apply.</li> <li>• Uses your default printer in Windows or the label printer you select in <b>Tools &gt; Options &gt; Printing</b>.</li> </ul> <p>See Cataloging/"Print labels" for more information about how the client produces labels.</p>
<p><b>Return values</b></p>	<p>TRUE if the action is successful, or FALSE if not.</p>
<p><b>Example</b></p>	<pre> Sub Main()      Dim CS As Object     Dim sSpine as String     Dim sCallNumber1 as String     Dim sAuthor1 as String     Dim sTitle1 as String     Dim sCallNumber2 as String     Dim sAuthor2 as String     Dim sTitle2 as String      Set CS = CreateObject("Connex.Client")     If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "gon,wi,th,w"      CS.GetLabelSpine "ZMYA", sSpine     CS.GetLabelCallNumber "ZMYA", sCallNumber1     CS.GetLabelAuthor(sAuthor1)     CS.GetLabelTitle(sTitle1)      sCallNumber2 = sCallNumber1     sAuthor2 = sAuthor1         </pre>

	<pre>sTitle2 = sTitle1  'Edit label fields as required here batchStarted = CS.StartLabelBatch  If batchStarted = True Then     CS.PrintLabelEx "ZMYA", sSpine, sCallNumber1, sAuthor1, sTitle1,     sCallNumber2, sAuthor2, sTitle2, 2, "Part", "1,2", "Copy", "1,2"     CS.EndLabelBatch End If  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Starts a label batch print and prints two copies of labels for record number 1 for holding library code ZMYA, based on spine, call number, author, and title text you edited, with the following print constants:             <ul style="list-style-type: none"> <li>○ <i>Part 1 Copy 1</i></li> <li>○ <i>Part 1 Copy 2</i></li> <li>○ <i>Part 2 Copy 1</i></li> <li>○ <i>Part 2 Copy 2</i></li> </ul> </li> </ul>

## GetLabelSpine

<b>Syntax</b>	BOOL = CS.GetLabelSpine (sHoldingCode, sSpine)
<b>Use to</b>	Returns the spine text based on the holding library code as it would appear in a label for the current record, so that you can modify it. Same as using <b>View &gt; Label</b> to open a label for the current record for editing.
<b>Parameters</b>	For <b>sHoldingCode</b> , enter the holding library code used to print the label, enclosed in quotation marks (determines the input stamps for the label). Enter the parameter <b>sSpine</b> to return the text of the label spine for editing.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sSpine as String     Dim nResult      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nResult = CS.GetLabelSpine ("ZMYA", sSpine)</pre>

	End Sub
<b>What this example does</b>	<ul style="list-style-type: none"> <li>○ Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>○ Searches WorldCat for record number 1.</li> <li>○ Returns the text of the spine label for editing.</li> </ul>

## GetLabelSpineUnformatted

<b>Syntax</b>	BOOL = CS.GetLabelSpineUnformatted (sHoldingCode, sSpine)
<b>Use to</b>	Returns unformatted text (no line breaks) for a spine label extracted from the current record, based on the holding library code, so that you can modify it. Same as using <b>View &gt; Label</b> to open a label for the current record for editing, except that the label text has no line breaks..
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>sHoldingCode</b>, enter the holding library code used to print the label, enclosed in quotation marks (determines the input stamps for the label).</li> <li>• Enter the parameter <b>sSpine</b> to return the text of the label spine for editing.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sSpine as String     Dim nResult      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nResult = CS.GetLabelSpineUnformatted ("ZMYA", sSpine)  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Returns the text of the spine label unformatted (no line breaks) for editing.</li> </ul>

## GetLabelCallNumber

<b>Syntax</b>	BOOL = CS.GetLabelCallNumber (sHoldingCode, sCallNumber)
<b>Use to</b>	Returns the call number based on the holding library code as it would appear in a label for the current record, so that you can modify it. Same as using <b>View &gt; Label</b> to open a label for the current record for editing.

<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sHoldingCode</b>, enter the holding library code used to print the label, enclosed in quotation marks (determines the input stamps for the label).</li> <li>Enter the parameter <b>sCallNumber</b> to return the call number for editing.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sCallNumber as String     Dim nResult      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nResult = CS.GetLabelCallNumber ("ZMYA", sCallNumber)  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>Searches WorldCat for record number 1.</li> <li>Returns the call number for editing.</li> </ul>

## GetLabelAuthor

<b>Syntax</b>	BOOL = CS.GetLabelAuthor (sAuthor)
<b>Use to</b>	Returns the author text from the current record, so that you can modify it. Same as using <b>View &gt; Label</b> to open a label for the current record for editing.
<b>Parameters</b>	Enter the parameter <b>sLabelAuthor</b> to return the author for editing.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sAuthor as String     Dim nResult      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nResult = CS.GetLabelAuthor (sAuthor)</pre>

	End Sub
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Returns the author text for editing.</li> </ul>

## GetLabelText

<b>Syntax</b>	BOOL = CS.GetLabelText (sTitle)
<b>Use to</b>	Returns the title text from the current record, so that you can modify it. Same as using <b>View &gt; Label</b> to open a label for the current record for editing.
<b>Parameters</b>	Enter the parameter <b>sTitle</b> to return the title for editing.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Dim sTitle as String     Dim nResult      Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      nResult = CS.GetLabelText (sTitle)  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Returns the title text for editing.</li> </ul>

## StartLabelBatch

<b>Syntax</b>	BOOL = CS.StartLabelBatch
<b>Use to</b>	Starts the label printing process.
<b>Parameters</b>	None
<b>Comments</b>	You must use this command to start label printing before you can use the <b>PrintLabel</b> or <b>PrintLabelEx</b> command in a macro.
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	See the example for <b>PrintLabel</b> or <b>PrintLabelEx</b> .

## EndLabelBatch

<b>Syntax</b>	BOOL = CS.EndLabelBatch
<b>Use to</b>	Ends the label printing process.
<b>Parameters</b>	None
<b>Comments</b>	You must use this command to end label printing after you use the <b>PrintLabel</b> or <b>PrintLabelEx</b> command in a macro.
<b>Return values</b>	None
<b>Example</b>	See the example for <b>PrintLabel</b> or <b>PrintLabelEx</b> .

## PrintMacro

<b>Syntax</b>	BOOL = CS.PrintMacro (sMacroName)
<b>Use to</b>	Print a specified macro.
<b>Parameters</b>	For <b>sMacroName</b> , enter the name of the macro, enclosed in quotation marks, in the following format (do not type the brackets): <b>[macro book name]![macro name]</b>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()     Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.PrintMacro "ClientMacros.mbk!Test" End Sub</pre>
<b>What this example does</b>	Prints a macro named <i>Test</i> in a macro book file named <i>ClientMacros</i> (always has the file extension <i>.mbk</i> )

## RunMacro

<b>Syntax</b>	BOOL = CS.RunMacro (sMacroName)
<b>Use to</b>	Run a specified macro.
<b>Parameters</b>	For <b>sMacroName</b> , enter the name of the macro, enclosed in quotation marks, in the following format (do not type the brackets): <b>[macro book name]![macro name]</b>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()     Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.RunMacro "ClientMacros.mbk!Test" End Sub</pre>
<b>What this example does</b>	Runs a macro named <i>Test</i> in a macro book named <i>ClientMacros</i> (always has the file extension <i>*.mbk</i> ).

## AssignMacroToKey

<b>Syntax</b>	BOOL = CS.AssignMacroToKey (sKeyName, sMacroName, bOverwrite)
<b>Use to</b>	Assign a keystroke shortcut to a specified macro.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sKeyName</b>, enter a key name enclosed in quotation marks.</li> <li>For <b>sMacroName</b>, enter the name of the macro to which you want to assign the key. Enclose the name in quotation marks. Enter in the following format (do not type the brackets):</li> <li><b>[macro book name]![macro name]</b></li> <li>(Optional.) For <b>bOverwrite</b>, enter <b>true</b> to override any existing assignment to the key. Otherwise, the new assignment is ignored and the existing assignment is retained.</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()     Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.AssignMacroToKey "F10", "ClientMacros.mbk!Test", True End Sub</pre>
<b>What this example does</b>	Assigns the F10 key to run a macro named <i>Test</i> in a macro book named <i>ClientMacros</i> (always has the file extension *.mbk). The key assignment overrides any existing assignment for <F10>.

## QueryRecordStatus

<b>Syntax</b>	BOOL = CS.QueryRecordStatus (sStatus, sReturnValue)
<b>Use to</b>	Request the specified status of the current record.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sStatus</b>, enter the type of status* you want to know about (use uppercase).</li> <li>Enter the parameter <b>sReturnValue*</b> to store the status of the current record for the specified status type.</li> </ul> <p>*See the list of values for available status types below.</p>
<b>Return values</b>	TRUE if successful, or FALSE if not or if the current record does not support the status.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object Dim sValue As String     Set CS = CreateObject("Connex.Client")      If CS.IsOnline = False Then         CS.Logon "", "", ""     End If      CS.Search "WC", "#1"      CS.QueryRecordStatus "LOCK", sValue      MsgBox sValue     CS.QueryRecordStatus "NUM", sValue</pre>

	<pre> MsgBox sValue  CS.QueryRecordStatus "TOTAL", sValue  MsgBox sValue End Sub </pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Logs on to Connexion if not already logged on, using the default authorization and password you selected in <b>Tools &gt; Options &gt; Authorizations tab</b>.</li> <li>• Searches WorldCat for record number 1.</li> <li>• Requests and returns the lock status of the record.</li> <li>• Requests and returns the record's control number.</li> <li>• Requests and returns the total number of holdings for the record.</li> </ul>

**Available status types and values**

Status type (enter in uppercase)	Definition and type of record that supports the status	Possible return values
NUM	Control number	1 through 99999999
SAVE	Bibliographic and authority Online or local save file number	1 through 9999
VAL	Bibliographic and authority Validation status	c (completed), f (failed), r (ready), - (not set)
UPD	Bibliographic and authority Update status	c (completed), f (failed), r (ready), - (not set)
PRD	Bibliographic Produce status	c (completed), f (failed), r (ready), - (not set)
DELH	Bibliographic Delete Holdings status	c (completed), f (failed), r (ready), - (not set)
XPO	Bibliographic Export status	c (completed), f (failed), r (ready), - (not set)
LBL	Bibliographic and authority Label status	c (completed), f (failed), r (ready), - (not set)
ADD	Bibliographic Add to Authority File status Authority	c (completed), f (failed), r (ready), - (not set)

Status type (enter in uppercase)	Definition and type of record that supports the status	Possible return values
LOCK	Lock status	y (yes), n (no)
RPL	Bibliographic and authority Replace status	c (completed), f (failed), r (ready), - (not set)
SOURCE	Bibliographic and authority Source of the record	Derived, Extracted, Imported, OCLC, URL, Workform
EDIT	Bibliographic Edit	y (yes), n (no)
HELD	Bibliographic and authority Holding status	y (yes), n (no)
WORKFLOW	Bibliographic Workflow status	Completes, In Process, In Review, New
MYSTATUS	Bibliographic and authority My Status	My Status you entered in free text (limit: 40 characters, not case-sensitive)
TOTAL	Bibliographic and authority Total number of holdings  Bibliographic	A number representing the total number of holdings

## LoadProfile

<b>Syntax</b>	BOOL = CS.LoadProfile (sName)
<b>Use to</b>	Load or create a user profile
<b>Parameters</b>	<ul style="list-style-type: none"> <li>For <b>sName</b>, enter the name assigned to the user profile you want to load, enclosed in quotation marks.</li> <li>Or</li> <li>Enter a name to create a new user profile based on client default settings.</li> </ul>
<b>Comments</b>	<ul style="list-style-type: none"> <li>LoadProfile is for users who create multiple profiles to catalog for different libraries using different sets of user options, local files, settings, etc.</li> <li>The client installs with one (default) profile named <i>MyProfile</i>.</li> <li>User profiles are listed in the Profile Manager window (<b>Tools &gt; Profiles</b>).</li> </ul>
<b>Return values</b>	TRUE if the action is successful, or FALSE if not.
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      If CS.LoadProfile "HealthSci"         MsgBox "Profile HealthSci loaded"     End If</pre>

	End Sub
<b>What this example does</b>	<ul style="list-style-type: none"> <li>• Sets the <i>HealthSci</i> profile as the default and loads the profile. Or If a <i>HealthSci</i> profile does not exist, creates a new profile with that name using system default settings, sets the new profile as the default, and loads it.</li> <li>• If you have local files open, closes them. If you are logged on, logs you off.</li> </ul>

## SendKeys

<b>Syntax</b>	CS.SendKeys string\$ [, wait%]
<b>Use to</b>	Send keystrokes to the client.
<b>Parameters</b>	<ul style="list-style-type: none"> <li>• For <b>string\$</b>, enter an expression containing the characters to send. See <b>Comments</b> for how to represent keystrokes with characters.</li> <li>• For <b>[, wait%]</b>, enter a numeric expression to determine whether to wait until all keys are processed before continuing program execution: <ul style="list-style-type: none"> <li>○ -1 = wait</li> <li>○ 0 = do not wait (default)</li> </ul> Both parameters are required. </li> </ul>
<b>Recommendation</b>	<p>OCLC recommends that you precede a CS.SendKeys string in a macro with an empty CS.SendKeys string to help stabilize the way SendKeys works.</p> <p><b>Example:</b> Dim CS As Object Set CS = CreateObject ("Connex.Client") CS.SendKeys "", -1 CS.SendKeys "{()abc{}}", -1Sends the keys (<i>abc</i>).</p>
<b>Comments</b>	<ul style="list-style-type: none"> <li>• To specify an ordinary character, enter it in the string. For example, to send character 'a' enter "a" as string. Several characters can be combined in one string: string "abc" means send 'a', 'b', and 'c'.</li> <li>• To specify that Shift, Alt, or Control keys should be pressed simultaneously with a character, prefix the character with: <ul style="list-style-type: none"> <li>○ + to specify Shift</li> <li>○ % to specify Alt</li> <li>○ ^ to specify Control</li> </ul> </li> <li>• Parentheses can be entered to specify that the Shift, Alt, or Control key should be pressed with a group of characters. For example, "%(abc)" is equivalent to "%a%b%c".</li> <li>• Since '+', '%', '^', '(' and ')' characters have special meaning to SendKeys, they must be enclosed in braces if they need to be sent with SendKeys. For example, <b>string "{%}"</b> specifies a percent character '%'. </li> <li>• The other characters that need to be enclosed in braces are '~' which stands for a new line, "Enter" if used by itself, and braces themselves: enter <b>{}</b> to send '{' and <b>{}}</b> to send '}'. Brackets '[' and ']' do not have special meaning to SendKeys but might have special meaning in other applications; therefore, they also need to be enclosed inside braces.</li> <li>• To specify that a key needs to be sent several times, enclose the character in braces and specify the number of keys sent after a space. <b>Example:</b> Enter <b>{X 20}</b> to send 20 'X' characters.</li> </ul>

	<ul style="list-style-type: none"> <li>To send one of the non-printable keys, enter a special keyword inside braces, as shown in the list below.</li> <li>To send one of function keys F1 through F12, enclose the name of the key inside braces. For example, to send the F5 key, enter "{F5}".</li> <li>Note that special keywords can be entered in combination with +, %, and ^. For example: %<b>{TAB}</b> represents <i>Alt-Tab</i>. Also, you can send several special keys in the same way as you would send several normal keys: <b>{UP 25}</b> sends 25 Up arrows.</li> <li>SendKeys can send keystrokes only to the currently active application. Therefore, you have to use the AppActivate statement to activate an application before sending keys (unless it is already active). (See OML Help for more information about AppActivate.)</li> <li>SendKeys cannot be used to send keys to an application that was not designed to run under Windows.</li> </ul>
<b>Return values</b>	None
<b>Example</b>	<pre>Sub Main()      Dim CS As Object     Set CS = CreateObject("Connex.Client")      CS.SendKeys "", -1     CS.SendKeys "{F1}", 0  End Sub</pre>
<b>What this example does</b>	<ul style="list-style-type: none"> <li>Sends an empty SendKeys string (see recommendation above).</li> <li>Sends the F1 function key to the client to open Help. (F1 is the default keystroke assigned to open Connexion Client Help).</li> </ul>

**Keywords used to send non-printable keys**

Key	Keyword
Backspace	{BACKSPACE} or {BKSP} or {BS}
Break	{BREAK}
Caps Lock	{CAPSLOCK}
Clear	{CLEAR}
Delete	{DELETE} or {DEL}
Down Arrow	{DOWN}
End	{END}
Enter	{ENTER}
Esc	{ESCAPE} or {ESC}
Help	{HELP}
Home	{HOME}
Insert	{INSERT}
Left Arrow	{LEFT}
Num Lock	{NUMLOCK}
Page Down	{PGDN}
Page Up	{PGUP}

Key	Keyword
-----	---------

Right Arrow	{RIGHT}
Scroll Lock	{SCROLLLOCK}
Tab	{TAB}
Up Arrow	{UP}

## 12. ASCII and hexadecimal codes for diacritics and special characters

### Enter diacritics and special characters using macros

If data used with macro commands such as CS.GetField or CS.SetField contains diacritics or special characters, the data appears with diacritics following the letters to which they belong. They do not appear as Unicode characters.

The standard function Asc returns the ASCII code mapped to the character. See an example of inserting characters in the description of ControlHeadingsAll in “Connexion client macro commands: Edit records.”

The following table gives the ASCII codes and the equivalent MARC-8 hexadecimal codes for diacritics and special characters available for use when you write macros that insert characters (the client inserts the correct information automatically when you record macros):

Character	ASCII decimal code	MARC-8 hexadecimal code
Acute	226	E2
ae, lowercase	181	B5
AE, uppercase	165	A5
Alif	174	AE
Ayn	176	B0
Breve	230	E6
British pound	185	B9
Candrabindu	239	EF
Cedilla	240	F0
Circle above letter	234	EA
Circle below letter	244	F4
Circumflex, non-spacing	227	E3
d with crossbar, lowercase	179	B3
D with crossbar, uppercase	163 (same as eth, uppercase)	A3
Degree sign	Insert the following value for the ALA font code: 158	
Delimiter (also listed under Subfield delimiter)	223	DF
Dot below letter	242	F2
Dot in center of line	168	A8
Double acute	238	EE
Double dot below letter	243	F3
Double tilde (left half)	250	FA
Double tilde (right half)	251	FB
Double underscore	245	F5
Eszett	Insert the following value for the ALA font code: 159	
Eth, lowercase	186	BA
Eth, uppercase	163 (same as D with crossbar, uppercase)	A3

Character	ASCII decimal code	MARC-8 hexadecimal code
Euro	Insert the following value for the ALA font code: 160	
Fill character	252	FC
Grave, non-spacing	225	E1
Grave, spacing	96	60
Hacek	233	E9
High comma, centered	254	FE
High comma, off center	237	ED
Inverted cedilla	248	F8
Inverted exclamation point	201	C6
Inverted question mark	200	C5
l with slash, lowercase	177	B1
L with slash, uppercase	161	A1
Left hook	247	F7
Ligature (left half)	235	EB
Ligature (right half)	236	EC
Macron	229	E5
Miagkii znak	167	E7
Musical flat	169	A9
Musical sharp	204	C4
o with hook, lowercase	188	BC
O with hook, uppercase	172	AC
o with slash, lowercase	178	B2
O with slash, uppercase	162	A2
oe, lowercase	182	B6
OE, uppercase	166	A6
Phonogram copyright mark	203	C2
Plus/minus	171	AB
Pseudo question mark	224	E0
Right hook	241	F1
Script lowercase L	190	C1
Subfield delimiter	223	DF
Subscript 0	144	30
Subscript 1	145	31
Subscript 2	146	32
Subscript 3	147	33
Subscript 4	148	34
Subscript 5	149	35
Subscript 6	150	36
Subscript 7	151	37
Subscript 8	152	38
Subscript 9	153	39
Subscript left parenthesis	154	28
Subscript minus	156	2D
Subscript plus	157	2B
Subscript patent mark	170	AA

Character	ASCII decimal code	MARC-8 hexadecimal code
Subscript right parenthesis	155	29
Superior dot	231	E7
Superscript 0	128	30
Superscript 1	129	31
Superscript 2	130	32
Superscript 3	131	33
Superscript 4	132	34
Superscript 5	133	35
Superscript 6	134	36
Superscript 7	135	37
Superscript 8	136	38
Superscript 9	137	39
Superscript left parenthesis	138	28
Superscript minus	141	2D
Superscript plus	140	2B
Superscript right parenthesis	139	29
Thorn, lowercase	180	B6
Thorn, uppercase	164	A4
Tilde, non-spacing	228	E4
Tilde, spacing (use in URLs)	126	7E
Turkish i without dot, lowercase	184	B8
Tverdyi znak	183	B7
u with hook, lowercase	189	BD
U with hook, uppercase	173	AD
Umlaut	232	EB
Underscore, non-spacing	246	F6
Underscore, spacing (use in URLs)	95	5F
Upadhamaniya	249	F9