OpenSpan Elective Training

OpenSpan Studio - Text Adapters

August 2010

OpenSpan, Inc.
4501 North Point Parkway
Suite 140
Alpharetta, Georgia 30022

Contents

This page left intentionally blank.

INTRODUCTION

OpenSpan Studio provides adapters to integrate with terminal emulator applications. OpenSpan Text adapters enable you to create projects which integrate mainframe applications. OpenSpan provides Text adapter templates for the most commonly used emulator software, including:

- Attachmate
- Ericom
- IBM wIntegrate

OpenSpan supports emulators which use WinHllapi, EHllapi, and some COM technologies.

In this training module, you will learn how to use the Windows adapter to integrate a standard WinHllapi emulator (Attachmate myEXTRA! Enterprise) and the Generic WinHllapi adapter to interact with a mainframe session running in myEXTRA! Enterprise. By the end of this training module, you will be able to:

- Configure the Windows adapter to integrate a terminal emulator application
- Configure and use the Generic WinHllapi adapter to integrate screens and fields from a mainframe application
- Interrogate screens, fields, and regions within a mainframe application
- Create automations to navigate a mainframe application and return data from the application

Prerequisites

This course requires successful completion of the **OpenSpan Studio Basics** training module. You will need a working knowledge of OpenSpan Studio and the ability to create basic Windows based projects.

The course requires the following software:

- OpenSpan Studio 4.5 or OpenSpan Plug-In for Microsoft Visual Studio
- Attachmate myEXTRA! Enterprise version 8.0
- Trace Player application
- UMich_large.BAT trace file for use with the TracePlayer (this file simulates an IBM-3278 terminal connected to the University of Michigan Libraries Databases)

Conventions

You can save time using this training guide by understanding how screen elements, input data, and definitions are shown.

Convention	Meaning
Black bold characters	**Names of program elements, such as buttons, commands, and dialog boxes, are shown in black bold text.**
Blue Bold Characters	**Text that you are supposed to type or data selections, such as from drop-lists, appear in blue boldface characters.**
Remember	**Definitions of terms and important concepts that bear remembering.**

CHAPTER 1: USING WINDOWS ADAPTER WITH EMULATOR APPLICATION

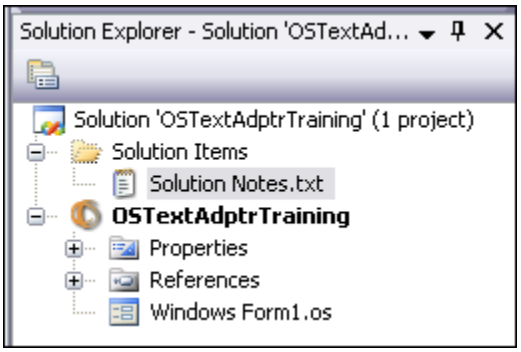
You can use the OpenSpan **Windows adapter** to control starting and stopping an emulator application. By using the Windows adapter along with the **Generic WinHllapi** text adapter, you can integrate any WinHllapi-based emulator application. For this course, the Attachmate myEXTRA! Enterprise emulator application is used. The role of the Windows adapter is to integrate with the emulator application. The host/mainframe application running in the emulator is integrated into the OpenSpan project using the Generic WinHllapi adapter.

In this chapter, you will create a new OpenSpan project containing a Windows adapter with the adapter properties set to integrate the Attachmate myEXTRA! Enterprise emulator application.

Begin by launching the OpenSpan Studio application. If you are using a VM provided by OpenSpan for this course, a base solution containing a Windows Form that will be used in throughout this course is included in the Projects folder (see the C:\Documents and Settings\user\My Documents\OpenSpan Studio for VS 2008\Project\OSTextAdptrTraining folder).

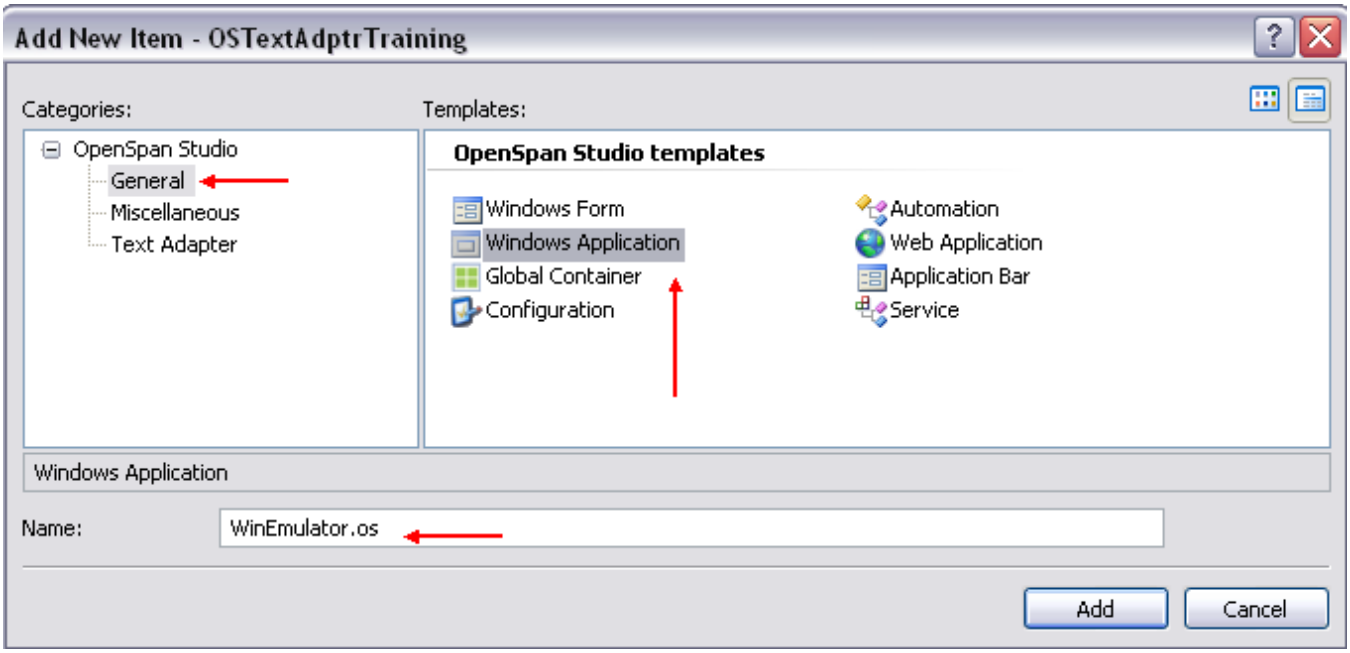
Create Project and add Windows Adapter

1. Select **File | Open Project/Solution**. The **Open Project** dialog displays.
2. Select the **OSTextAdptrTraining** folder and select the **OSTextAdptrTraining.ossln** solution. A solution containing a solution Text file (SolutionNotes.txt), OpenSpan project (OSTextAdptrTraining) and Windows Form1.os project item displays. An example of the **Solution Explorer** follows:

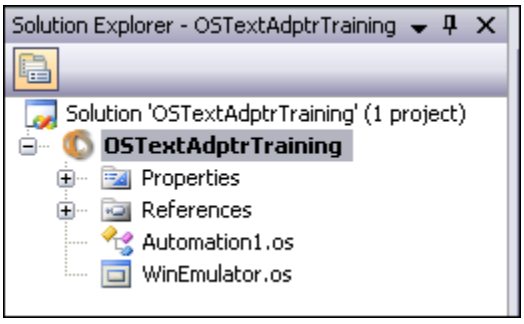


3. Right-click on the OpenSpan project and select Add **| New Windows Application** from the local menu. The **Add New Item** dialog displays.
4. Select the Windows Application template from the **OpenSpan Studio | General** category.

5. Name the adapter item **WinEmulator**. An example of the completed **Add New Item** dialog follows:



6. Click **Add** to add the Windows Application adapter project item and dismiss the Add New Item dialog. The adapter project item is added to the Solution Explorer:



7. Select **Save All** (from File menu or main toolbar) to save all solution files.

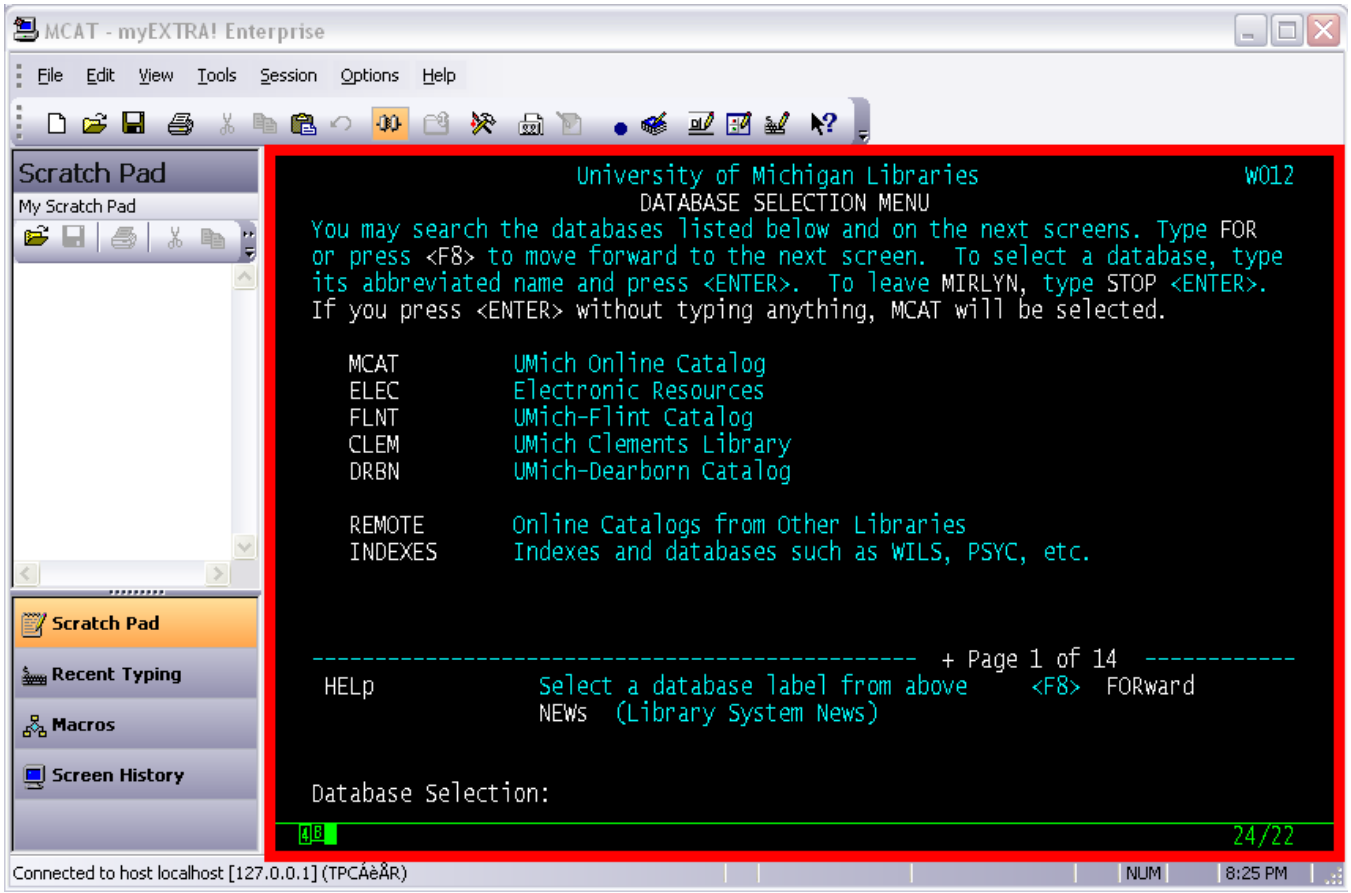
Set Windows Adapter Properties

In this sample solution, the Windows adapter integrates the emulator application into the project. Use the following steps to

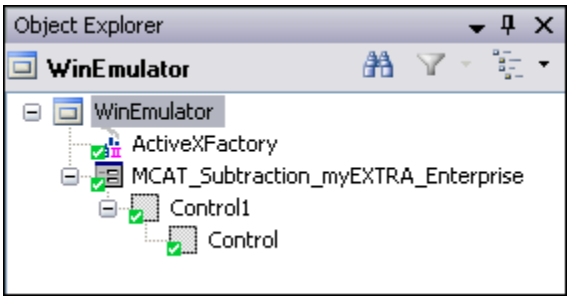
- Set the **Path** property to the emulator application name and path
- Interrogate the emulator application – session display window

1. Highlight the Windows adapter project item (WinEmulator.os) in the Solution Explorer and double-click the item to open it in the **Designer** (if not already open).
2. Open the **Properties** window.
3. Click in the **Path** property to display the Browse button.

4. Click the Browse button. The **Open** dialog displays.
5. Use the Open dialog to navigate to the location of the emulator application file (Extra.exe). If you are using a VM provided by OpenSpan for this course, the application is located in the C:\Program Files\Attachmate\E!E folder. In this case, set the Path property to: **C:\Program Files\Attachmate\E!E\EXTRA.exe**
6. In the **Arguments** property, enter the name of the emulator session for the project. If you are using a VM provided by OpenSpan for this course, the session is MCAT.edp and it is located in the C:\Documents and Settings\user\My Documents\Attachmate\EXTRA!\sessions folder. In this case, set the Arguments property to : **C:\Documents and Settings\user\My Documents\Attachmate\EXTRA!\sessions\MCAT.edp**
7. Leave default entries for the remaining Windows Adapter properties.
8. **Note**: If not auto-started in your VM or other environment, then before interrogation, you must start the batch file **UMich_large.BAT**. If manual start of the batch is necessary, and you are using a VM provided by OpenSpan for this course, you may already have a desktop shortcut that points to this file. Double-click either the shortcut or the batch icon itself, found in the path **C:\Traceplayer-1\UMich_large.BAT.** After the UMICH.edp session loads, you are ready to proceed.
9. Click the **Start Interrogation** button on the **Design** page for the Windows adapter. The **Interrogation Form** displays and the emulator application is launched. If you are using a VM provided by OpenSpan for this course, the MCAT session opens in the myEXTRA! Enterprise application.
10. Interrogate the session display area within the emulator application. See the following illustration:



11. The following controls are created and added to the Object Explorer:



12. Highlight Control (the child of Control1) and delete the ControlID match rule. This ID changes between application sessions and can cause matching errors. Repeat this process for Control1.
13. Stop the Interrogator by closing the Interrogation Form.
14. Save all solution files.

CHAPTER 2: USING GENERIC WINHLLAPI ADAPTER

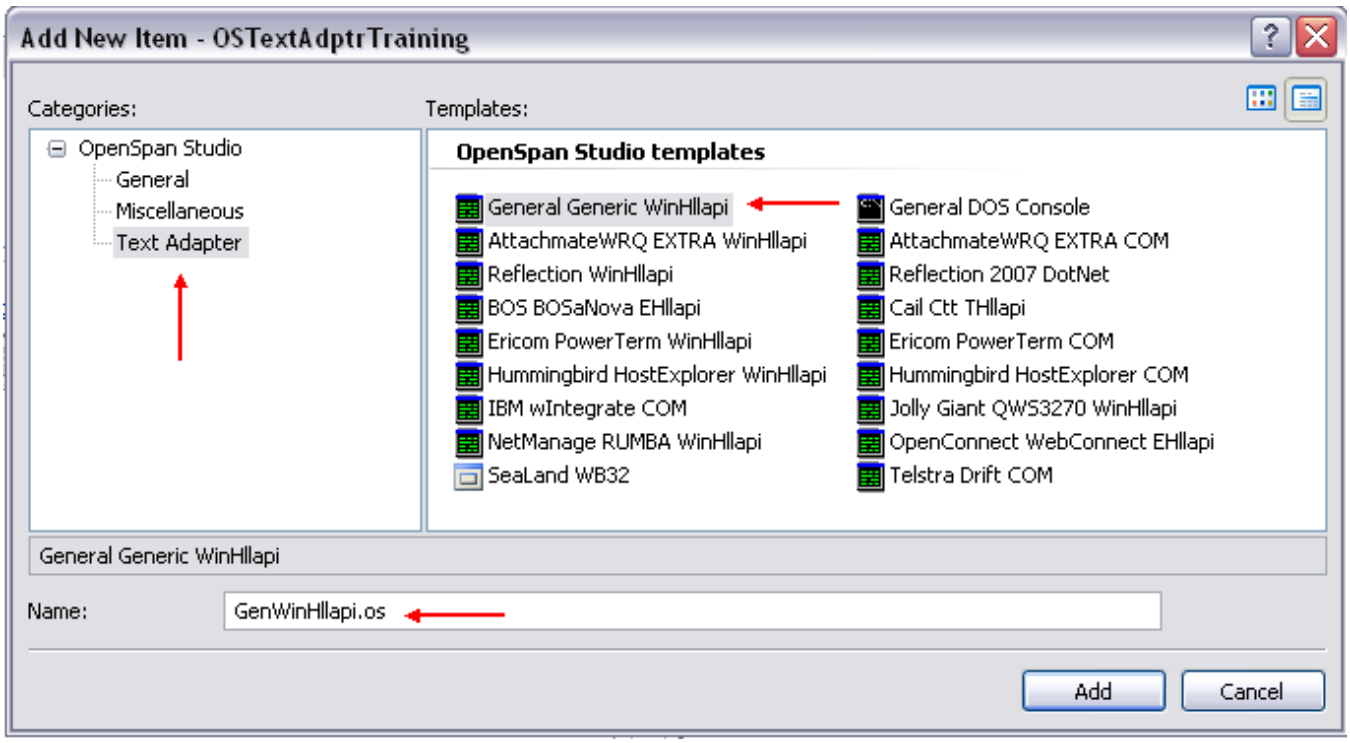
While OpenSpan provides adapters for specific emulators, such as the Attachmate EXTRA! WinHllapi, this course illustrates how to use the Generic WinHllapi adapter to integrate host applications running in any emulator which uses WinHllapi communications. By using the Generic adapter, you must provide logic in the project for starting the emulator and opening the session. In the previous chapter, you added the Windows adapter and configured it to start the myEXTRA! application and load the MCAT.edp session.

In this chapter, you will learn how to interrogate screens and screen text.

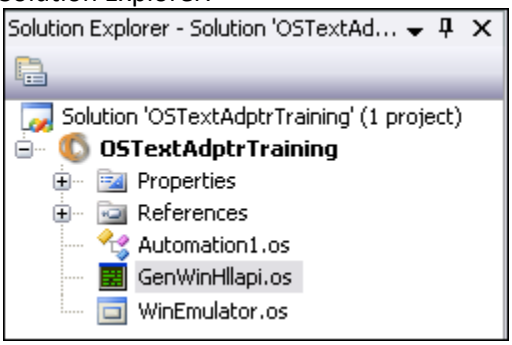
Begin by ensuring the Traceplayer application is running the UMICH_Large.bat file.

Add Generic WinHllapi Adapter

1. Right-click on the project, OSTextAdptrTraining in the Solution Explorer and select **Add | New Item** from the local menu. The Add New Item dialog displays.
2. Select the General Generic WinHllapi template from the OpenSpan Studio Text Adapter category.
3. Name the item: **GenWinHllapi.os**. An example of the completed New Item dialog follows:



4. Click Add to add the adapter item and dismiss the Add New Item dialog. The adapter item is added to the Solution Explorer:



5. Save all solution files.

Set Text Adapter Properties

The key Text Adapter properties used in basic text adapter projects are:

- **DllName** – sets the name and full path of the HLLAPI (High Level Language Application Program Interface) assembly used by the emulator. For Attachmate myEXTRA!, the emulator uses the WHLAPI32.dll assembly to communicate with mainframe applications.
- **SessionID** – identifies the HLLAPI Shortname that has been setup in the emulator application for the session. In Attachmate myEXTRA!, the HLLAPI Shortname is set on the Global Preferences dialog (Options | Global Preferences | Advanced tab).
- **StartOnProjectStart** – defines whether the adapter and associated session are started when the project starts. When this property is FALSE, you must add project logic to start the adapter (e.g., use the Adapter.Start() method).

1. Begin by highlighting the text adapter project item (GenWinHllapi.os) in the Solution Explorer.
2. Double-click on the text adapter to open the item in the Designer if not already open.
3. Open the Properties window and click in the DllName property to display a Browse button.
4. Click on the Browse button. The Open dialog displays. Use this dialog to select the HLLAPI assembly used by the adapter. If you are using a VM provided by OpenSpan for this course, the assembly is WHLAPI32.dll and it is located in the C:\Program Files\Attachmate\E!E folder. In this case, the DllName property should be set to: C:\Program Files\Attachmate\E!E\WHLAPI32.dll
5. Set the SessionID property. If you are using a VM provided by OpenSpan for this course, the session shortname is A therefore set the SessionID to A.
6. Set the StartOnProjectStart property to False. For this training solution, the adapter will be started based on creation of the session window Control interrogated using the Windows adapter.
7. Save all solution files.

Text Adapter Interrogation Function

When interrogating a mainframe (text-based) application, you define Screens, Regions, and Fields. OpenSpan uses the position, length, and text of selected screen areas to identify the object and match it. The steps to matching targets in a text application are to:

- Identify the Screen containing the text you want to include in the project
- Select target text as either a Field (screen area in which text can be typed) or a Region (display-only text area).
- Use Partial Screens to identify targets that are repeated within the Screen presentation. (See Chapter 3 for details on partial screens.)
- Modify match rules from Simple to Complex when you need to generalize the text to accommodate matching (see Chapter 3 for details on modifying match rules).
- In order to interrogate the session screens within the OpenSpan Designer, OpenSpan must start the emulator application. For this course, you will start the interrogation function for the Windows adapter (WinEmulator.os) and then start the interrogator for the text adapter (GenWinHllapi.os).

Unlike Windows and Web application interrogation, there is no interrogator icon to drop over targets. Instead, you left-click and drag the cursor through the screen text displaying in the OpenSpan Designer. This highlights the selected text. You then right-click on the highlighted text (or field area) to view the following interrogation options:

- Save As Match Region
- Select Field
- Save As Field
- Save As Region

Definitions of these options follow:

Save As Match Region

Use this function to select screen text that uniquely identifies a screen within the application. You can set a Match Region as either Simple or Complex. Simple match regions use the text exactly as it appears on the screen to match the screen. If you select Complex, you can use regular expression (regex) syntax to generalize the text for matching. This is particularly useful when the screen identifying text contains an account number or date which will change during runtime. You can use the Complex match rule options to match the screen text based on regex syntax. See Chapter 3 of this training module for more information on Complex match rules.

Select Field

Use this function to determine the field length of screen fields. Place the cursor in the screen area for the field, **right-click** and select this option. OpenSpan highlights the entire area defined by the host application as the field. This will assist you in determining whether you want to use the application's definition of a specific field or whether you will want to select a region (i.e., part of the area) to define a customized field.

Save As Field

Use this to select a field and create an OpenSpan control representing the field.

It is recommended that you use the Select Field option to highlight the entire field area and then use the Save As Field option to create a control for the field. This ensures that when data is entered in the field or retrieved from the field, the entire entry is recognized by OpenSpan.

Save As Region

Use this function to select areas of text for use in a project. In some cases the host application has defined a very large field length and therefore when you use the **Select Field** command a large area is highlighted. The **Save as Region** command enables you to limit this area or select a custom area of text as to return as a region to OpenSpan Studio.

Interrogating a Host Session

You interrogate host applications through the OpenSpan Designer. For this training module, the project will navigate the University of Michigan Libraries application to the MCAT database, query the database using a keyword search term, and retrieve book titles from the search results for display on a Windows form. The following screens and text areas are used in the training solution:

Screen - University of Michigan Libraries Database Selection Menu

- Database Selection field

Screen – UMich Online Catalog Introduction (Welcome to MCAT)

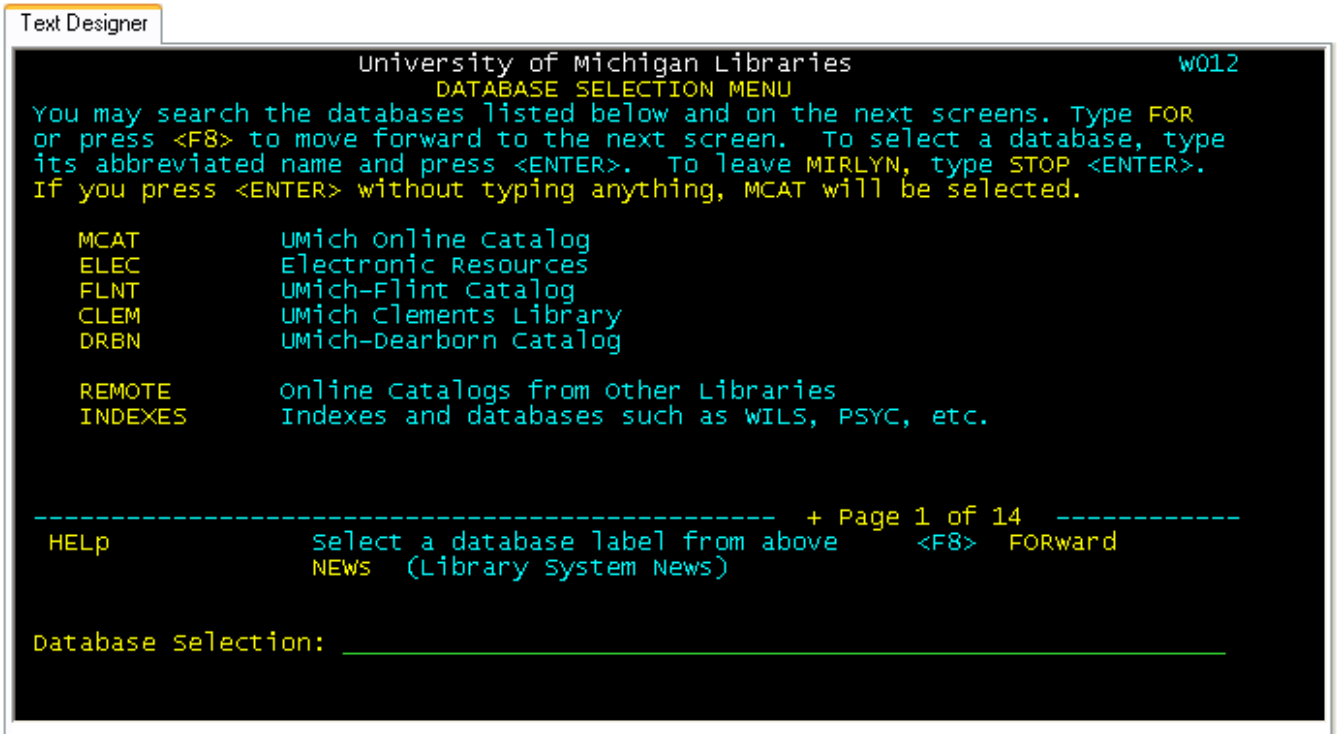
- Next Command field

Screen – UMich Online Catalog Keyword Index (Search Results screen)

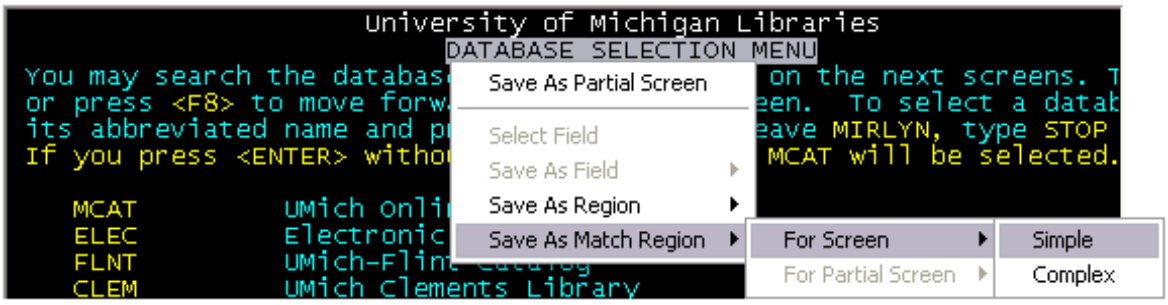
- Search Results field
- Result Records (Date, Title, Author)
- <F8> FORward page text

Before you begin, make sure the UMICH_Large.bat file is running. For more information on the Traceplayer, see Appendix A: Using the Traceplayer Application.

1. In OpenSpan Studio, start the emulator application by selecting the Windows adapter project item (WinEulator.os) and clicking the Start Interrogation button in the Designer. The emulator application launches and the UMICH.edp session loads.
2. Open the **Text Adapter** project item, GenWinHllapi.os, and select **Start Interrogation** from the **Designer**. **University of Michigan Libraries Database Selection Menu** screen is loaded into the OpenSpan Designer. Interrogate the session using the screen displayed in the Designer for the text adapter. An example of the Designer follows:

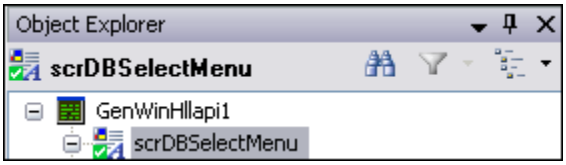


3. Identify the **Database Selection Menu** screen by highlighting the text in the OpenSpan Designer. **Left-click** just before the "D" of DATABASE and drag the cursor through the "Database Selection Menu" text string. OpenSpan highlights the text. Right-click on the highlighted text to display the interrogation options as follows:



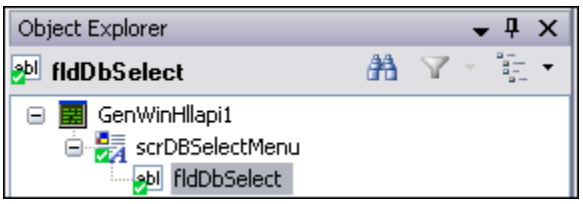
4. Select **Save as Match Region | For Screen | Simple** from the context menu. A Screen control is created and displays in the **Object Explorer**.

5. In the **Properties** window, change the **Name** property for the Screen to: **scrDBSelectMenu**. Your **Object Explorer** should look similar to the following:

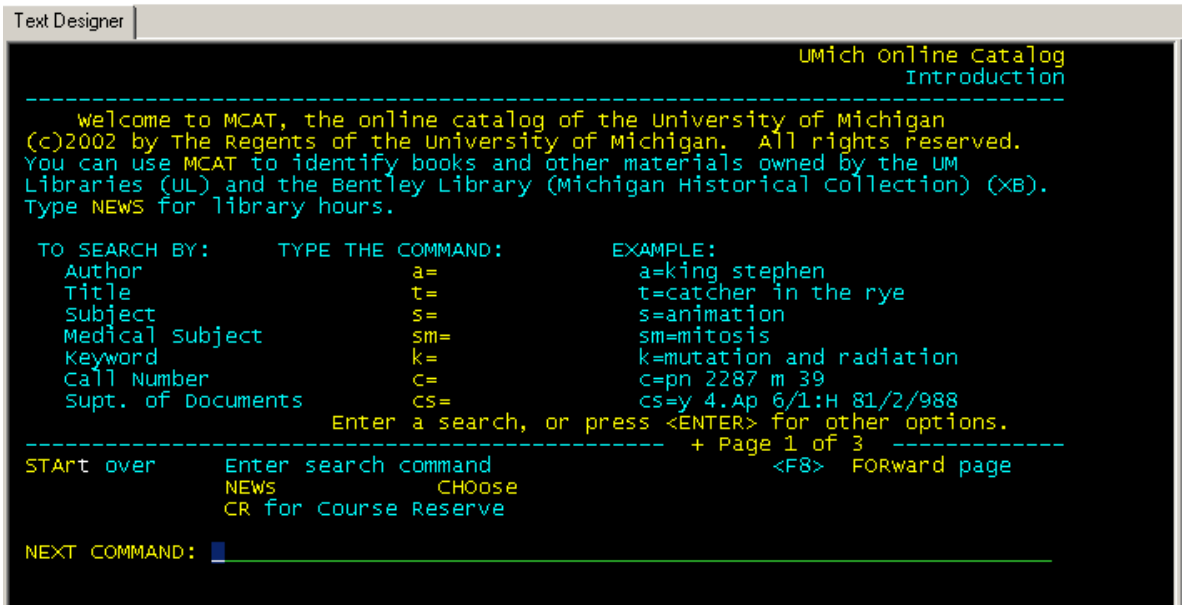


6. Interrogate the **Database Selection** field (the empty space just to the right of the Database Selection text). Right-click anywhere in the field and choose the **Select Field** option from the local menu. OpenSpan highlights the entire data entry field as follows:

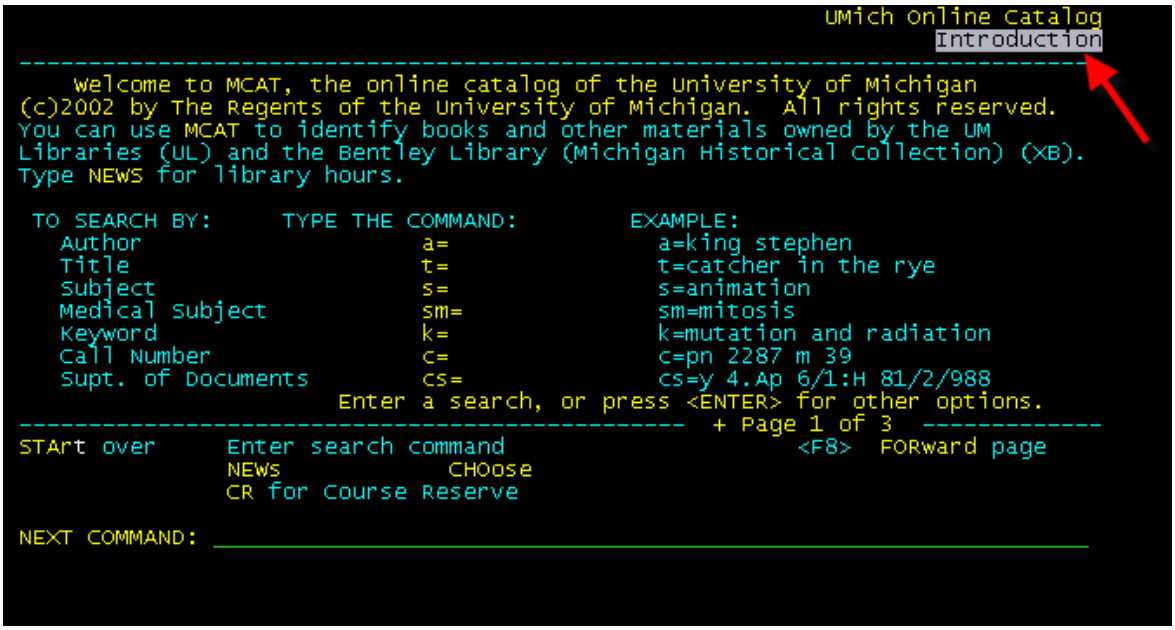
7. Right-click anywhere in the highlighted field and choose the **Save As Field | For Screen** option. A control for the field is added to the Object Explorer.
8. Use the Properties window to rename the control: **fldDbSelect**. Your Object Explorer should look like the following:



9. On the Database Selection Menu screen displayed in the emulator, enter MCAT in the Database Selection field and press Enter to navigate to the **UMich Online Catalog** screen. An example of this screen follows:

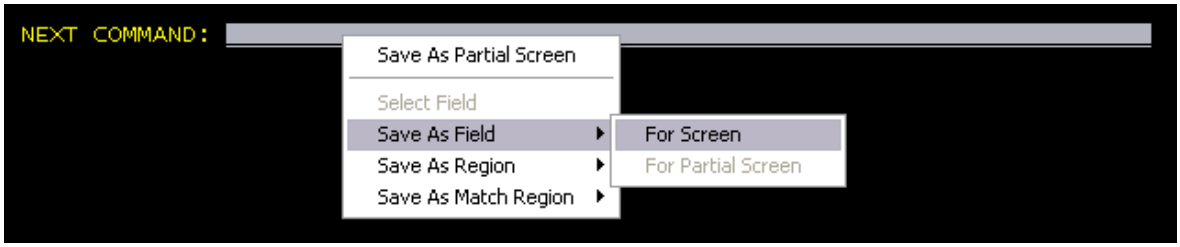


10. For the Welcome screen of the MCAT database, use the **Introduction** text to identify the screen by selecting (highlighting) the text **Introduction**. Then right-click on the selected text and choose **Save As Match Region | For Screen | Simple** from the local menu.

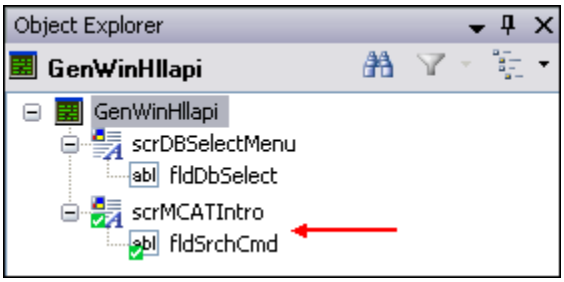


11. Next, right-click the **Next Command** data entry field (not the field label), and select **Select Field** from the context menu.

12. Right-click the highlighted **Next Command** data entry field and select **Save As Field | For Screen**.

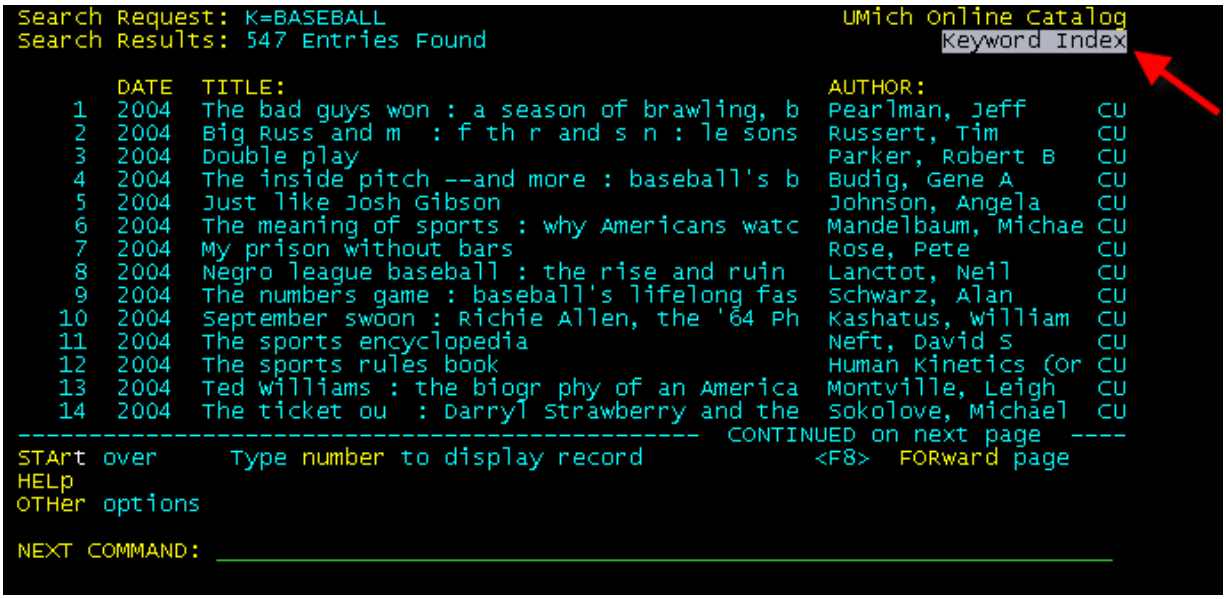


13. In the **Properties** window, rename **Screen1** to **scrMCATIntro** and **Field1** to **fldSrchCmd**. Your Object Explorer should look like the following:



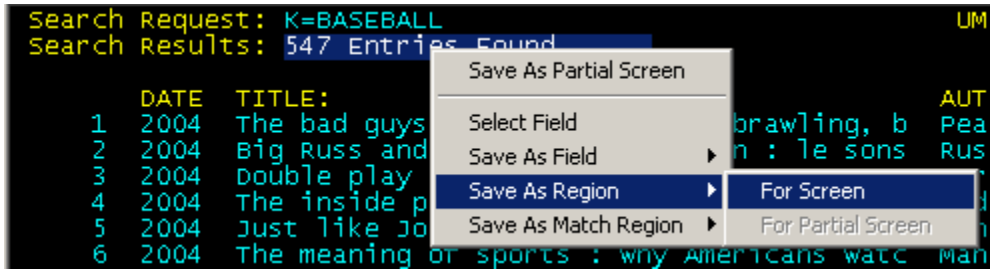
14. In the **Next Command** field on the **UMich Online Catalog- Introduction** screen (displayed in the OpenSpan Designer), enter the following command to search for books related to the keyword "baseball": **k=baseball**. Press **Enter** to display the search results. The results display in UMICH Online Catalog – Keyword Index screen as shown below:

15. Use the Keyword **Index** text to identify the **UMich Online Catalog Keyword Index** screen by selecting (highlighting) the label **Keyword Index.**

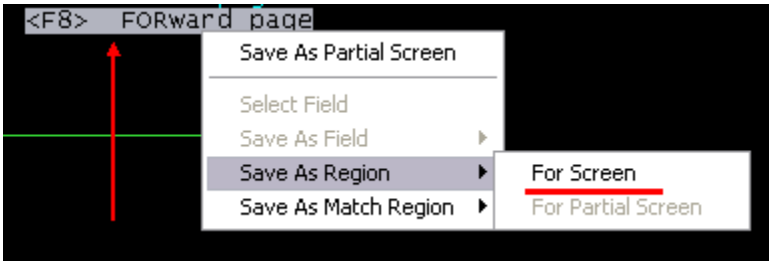


16. Then right-click the Keyword Index text and select **Save As Match Region | For Screen | Simple**.

17. Select (highlight) the **Search Results** region (i.e., **547 Entries Found**), leaving enough spaces to accommodate larger result sets. Right-click the highlighted region, and select **Save As Region | For Screen**.

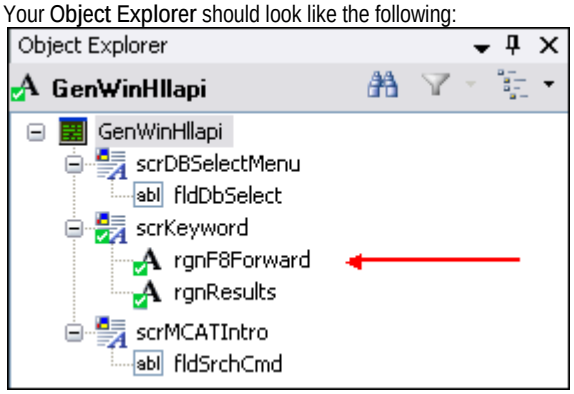


18. Highlight the text "<F8> FORward page" near the bottom of the screen. This text indicates that there are additional screens of search results. The project will use the text to determine whether additional records exist for the search. Right-click on the highlighted text and select **Save as Region | For Screen.**



19. In the **Properties** window, rename the interrogated controls:
    - **Screen1 -** **scrKeyword**
    - **Region 1** (search results) – **rgnResults**
    - **Region2** (F8 Forward) – **rgnF8Forward**

Your Object Explorer should look like the following:



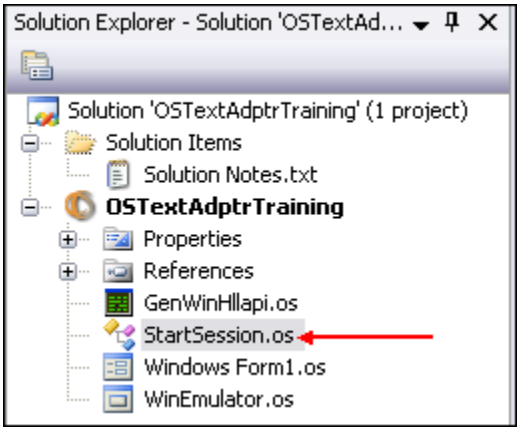
20. Click **Stop Interrogation** for the Text adapter (GenWinHllapi), then close the Interrogation Form for the Windows adapter, WinEmulator. When prompted, you can disconnect from the Attachmate session.
21. Save all solution files.

Add Automation to Start Text Adapter

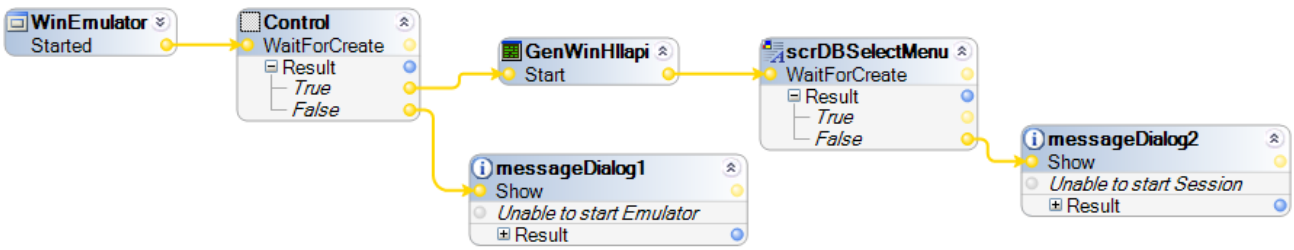
For this course, the Windows adapter will automatically launch the emulator application (Attachmate myEXTRA! Enterprise). Once the emulator is launched, the Text Adapter (GenWinHllapi.os) is started by calling the Start() method on the adapter. The first automation of this project starts the text adapter.

Begin by highlighting the OSTextAdptrTraining project in the Solution Explorer.

1. Right-click on the project and select **Add | New Automation** from the local menu. The Automation is added to the project and displays in the Solution Explorer and Object Explorer.
2. Select the Name property (in the Properties window) for the automation and rename it: **StartSession.os**. Your Solution Explorer should look similar to the following:



3. Add two OpenSpan messageDialog components to the automation – Local tab. Select the "(String message); DialogResult overload option for both components.
4. Set the message parameters for the dialogs as follows:
   - messageDialog1 – Unable to start Emulator. This message will display if OpenSpan in unable to start the windows emulator application (myEXTRA! Enterprise).
   - messageDialog2 – Unable to start Session. This message will display if OpenSpan is unable to load the UMICH session in the emulator through the text adapter.
5. Add the following properties, methods and events to the automation:
   - WinEmulator.Started event
   - WinEmulator.Control.WaitForCreate() method – this is the child control under Control1 for the Windows adapter.
   - GenWinHllapi.Start() method
   - ScrDBSelectMenu.WaitForCreate() method
   - messageDialog1.Show(message=Unable to start emulator) method
   - messageDialog1.Show(message=Unable to start session) method
6. Arrange the connection blocks and connect the event path as shown in the following image:



7. Save all solution files.

Build and Debug the Project

Make sure that the UMICH_Large.bat file is running and then do the following:

1. Select the Debug solution configuration from the Debug toolbar.
2. Click the **Debug** button. OpenSpan Compiles the project and launches a design version of the project in the OpenSpan Runtime application installed with OpenSpan Studio (or Plug-in).
3. The following project execution sequence should occur:
   a. The project is built and no build errors are reported.
   b. OpenSpan Runtime loads the design project OSTextAdptrTraining.
   c. OpenSpan Runtime loads the Search MCAT Topics windows form. Note that the search term list box is disabled. This listbox only becomes enabled when the MCAT query screen is displayed in the emulator.

d. OpenSpan Runtime launches the emulator (myEXTRA! Enterprise) and opens the UMICH.edp session.

e. The initial Database Selection Menu screen displays in the emulator.

4. Click the **Stop Debugging** button in OpenSpan Studio to unload the project and stop the OpenSpan Runtime application.

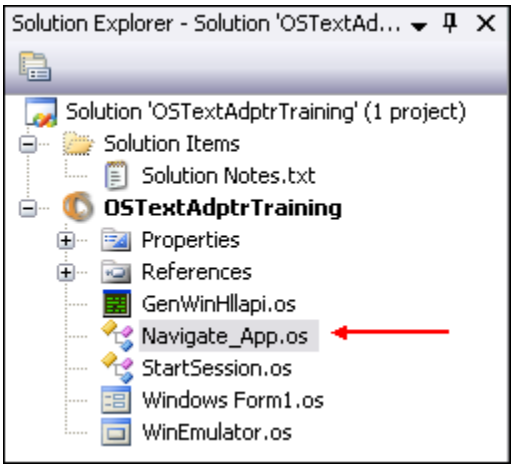
Add Automation to Navigate the host application screens

This automation will navigate the host application from the main screen, Database Selection Menu, to the MCAT database.

Begin by highlighting the OSTextAdptrTraining project in the Solution Explorer.

1. Right-click on the project and select **Add | New Automation** from the local menu. The Automation is added to the project and displays in the Solution Explorer and Object Explorer.

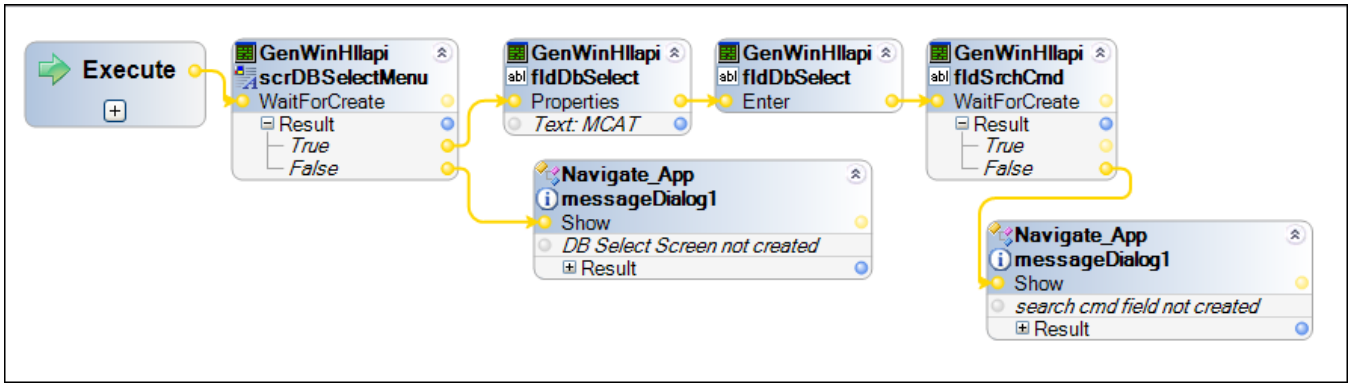
   Select the **Name** property (in the Properties window) for the automation and rename it: **Navigate_Apps.os**. Your Solution Explorer should look similar to the following:



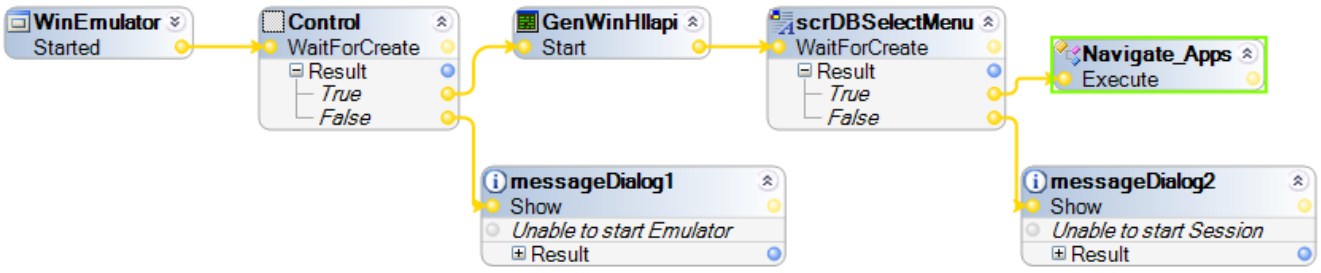
2. Add an OpenSpan messageDialog component to the automation – Local tab. Select the "(String message); DialogResult overload option.
3. Select the messageDialog1 component and copy/paste the component to add another instance of the messageDialog to the automation.
4. Set the message parameters for the dialogs as follows:
   - messageDialog1 – DB Select Screen Not Created. This message will display if OpenSpan in unable to match the Database Selection Menu screen.
   - messageDialog1 – Search Cmd field not created. This message will display if OpenSpan is unable to match the Search Command field on the MCAT Introduction screen indicating that the project was unable to navigate from the Database Selection screen to the MCAT Introduction screen.
5. Right-click on the automation and add an Entry point.
6. Add the following properties, methods and events to the automation:

- GenWinHllapi.scrDBSelectMenu.WaitForCreate() method
- GenWinHllapi.fldDbSelect.Text property – set the text parameter to MCAT.
- GenWinHllapi.fldDbSelect.Enter() method – this method executes pressing the Enter key in the Database Selection field after the MCAT text is set.
- GenWinHllapi.fldSrchCmd.WaitForcreate() method
- messageDialog1.Show(message=DB Select Screen not created) method
- messageDialog1.Show(message=Search Command field not created) method

7. Arrange the connection blocks and connect the event path as shown in the following image:



8. Save all solution files.
9. Return to the StartSession automation and add the Navigate_Apps.Execute() method. This method will cause the Navigate_Apps automation to run once the session has successfully started.
10. Connect the True output event node from the ScrDBSelectMenu.WaitForCreate method to the input event node of the Navigate_Apps.Execute() method. An example of the completed StartSession automation follows:



Build and Debug the Project

Make sure that the UMICH_Large.bat file is running and then do the following:

1. Select the Debug solution configuration from the Debug toolbar.
2. Click the **Debug** button. OpenSpan compiles the project and launches a design version of the project in the OpenSpan Runtime application installed with OpenSpan Studio (or Plug-in).
3. The following project execution sequence should occur:
    a. The project is built and no build errors are report.
    b. OpenSpan Runtime loads the design project OSTextAdptrTraining.
    c. OpenSpan Runtime loads the Search MCAT Topics windows form. Note that the search term list box is disabled. This listbox only becomes enabled when the MCAT query screen is displayed in the emulator.
    d. OpenSpan Runtime launches the emulator (myEXTRA! Enterprise) and opens the UMICH.edp session.
    e. The initial Database Selection Menu screen displays in the emulator.
    f. The text MCAT appears in the Select field and the host is navigated to the MCAT Introduction screen.
4. Click the **Stop Debugging** button in OpenSpan Studio to unload the project and stop the OpenSpan Runtime application.

Add Automation to Search MCAT Library

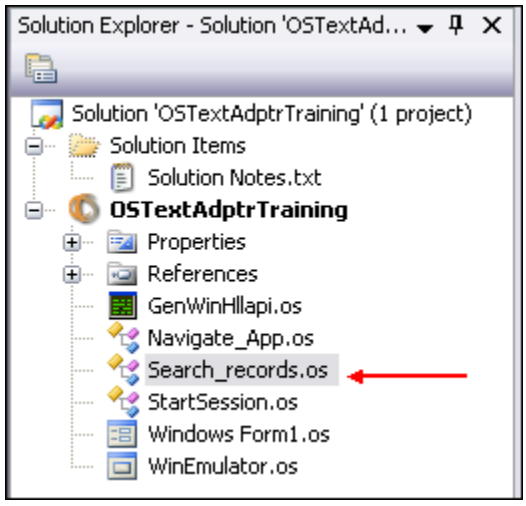
This automation will perform a keyword search based on the keyword selected on the Windows Form. The search begins when the user clicks the Search Selected Keyword button. Note that keyword listbox is disabled by default so that users cannot initiate a search unless the UMICH session is running and the UMICH Online Catalog Introduction screen is matched (including the Command field).

The automation has three parts:

- Enabling the Keyword listbox (lbxTerms) and clearing any existing Titles listed on the Windows Form.
- Searching for the keyword selected in the lbxTerms listbox.
- Clearing the form results (Titles and Number of Records).

Begin by highlighting the OSTextAdptrTraining project in the Solution Explorer.

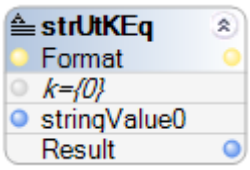
1. Right-click on the project and select **Add | New Automation** from the local menu. The Automation is added to the project and displays in the Solution Explorer and Object Explorer.
2. Select the Name property (in the Properties window) for the automation and rename it: **Search_records.os**. Your Solution Explorer should look similar to the following:



3. Add an OpenSpan messageDialog component to the automation – Local tab. Select the "(String message); DialogResult overload option. This message dialog will be used to display an error if the UMich Online Catalog Keyword Index (Search Results) screen is not matched.
4. Rename messageDialog1 component to: **msgKeywrdScrn**. Set the message parameter to: **Results Screen not Created**.
5. Add a StringUtils component to the automation – Local tab. Rename the component: **strUtKEq**. This component will be used to format the search command as required by the host application (e.g., k=baseball).
6. Right-click on the automation and add an Entry point. The Entry point will enable executing the automation only after the UMich Online Catalog Introduction (Welcome to MCAT), Next Command field is matched.
7. Add the following properties, methods and events to the automation:
   - WindowsForm1.BringToFront() method
   - WindowsForm1.lbxTerms.Enabled property – set the Enabled property to "TRUE" on the connection block.
   - WindowsForm1.lbxTitles.Items.Clear() method – add two instances of this method to the automation. To access this method, highlight the lbxTitles listbox in the Object Explorer and click the Explore Component Properties button. Next, highlight the Items property and use the Object Inspector to select the Clear method for the Items property. Return to the standard view of the Object Explorer by clicking on the Explore Components button.
   - WindowsForm1.btnSearch.Click event – this event initiates the search of MCAT library.
   - WindowsForm1.lbxTerms.SelectedItem property – this sets the keyword used in the search.
   - WindowsForm1.lblRecNo.Text property – add two instances of this property to the automation. The label displays the number of titles returned from the search.
   - WindowsForm1.btnClear.Click event – this event resets the results text shown on the Windows form.
   - GenWinHllapi.fldSrchCmd.Text property – sets keyword search command for MCAT.
   - GenWinHllapi.fldSrchCmd.Enter() method – simulates pressing Enter in the MCAT screen, search field.

- GenWinHllapi.scrKeyword.WaitForCreate() method – indicates when the search results have been displayed. **Add two instances of this method to the automation.**
- GenWinHllapi.rgnResults.Text property – displays the number of titles found for the search.
- msgKeywrdScrn.Show(Results Screen Not Created) method – reports an error if the UMich Online Catalog Keyword Index (Search Results screen), Results text region is not matched.

8. Select the strUtKEq stringUtils component in the Local tray and use the Object Inspector to select the Format(2 parameters – String formatString, String stringValue0) method. Add this method to the automation.
9. Set the formatString parameter for the strUtKEq.Format method to: k={0}.

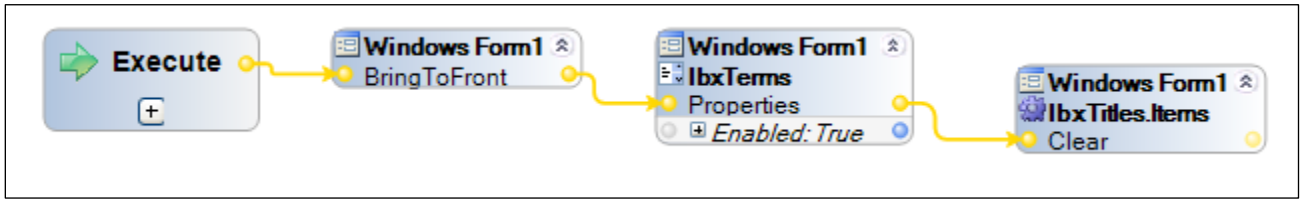


Note: Make sure to enter a zero (0) not the letter "O" inside the braces of the format string.

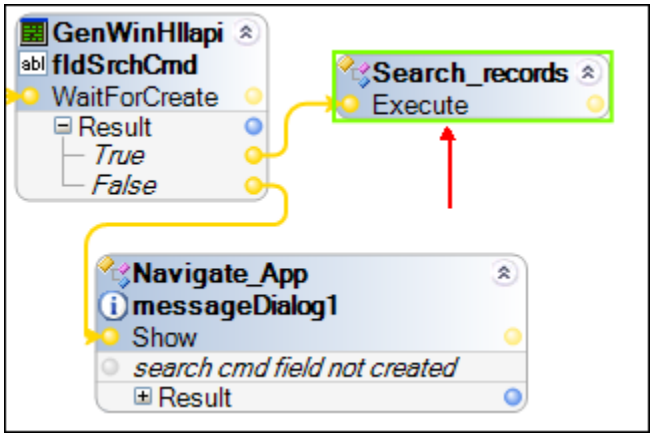
10. Save all solution files.

Establish Logic to Execute Automation and Enable Keyword Terms Listbox

1. On the Search_Records.os automation, arrange the connection blocks and connect the event path as shown in the following image:



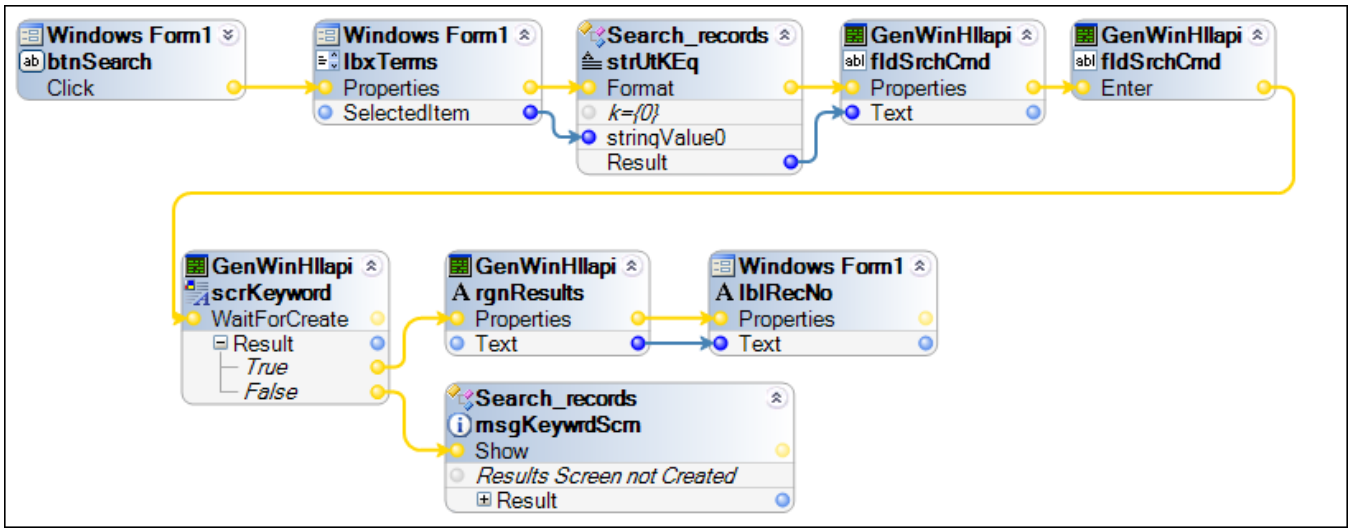
2. Open the Nagivate_Apps.os automation and add the Search_Records.Execute() method to the automation. Connect the event path for the method as follows:



3. Save all solution files.

Establish Logic to Execute Search

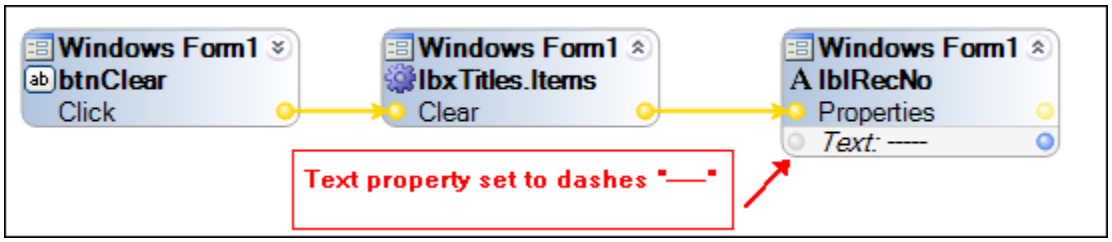
1. On the Search_Records.os automation, arrange the connection blocks and connect the event/data path as shown in the following image:



2. Save all solution files.

Establish Logic to Reset Windows Form for New Search

1. On the Search_Records.os automation, arrange the connection blocks and connect the event/data path as shown in the following image:



2. Click on the Text property of the lblRecNo label and set the text to dashes ("-----").
3. Save all solution files.

Build and Debug the Project

Make sure that the UMICH_Large.bat file is running and then do the following:

1. Select the Debug solution configuration from the Debug toolbar.
2. Click the Debug button. OpenSpan Compiles the project and launches a design version of the project in the OpenSpan Runtime application installed with OpenSpan Studio (or Plug-in).
3. The following project execution sequence should occur:

   a) The project is built and no build errors are reported.

b) OpenSpan Runtime loads the design project OSTextAdptrTraining.

c) OpenSpan Runtime loads the Search MCAT Topics windows form. Note that the search term list box is disabled. This listbox only becomes enabled when the MCAT query screen is displayed in the emulator.

d) OpenSpan Runtime launches the emulator (myEXTRA! Enterprise) and opens the UMICH.edp session.

e) The initial Database Selection Menu screen displays in the emulator.

f) The emulator navigates the host application to the UMich Online Catalog Introduction (Welcome to MCAT) screen.

g) The keyword listbox on the Windows form becomes enabled.

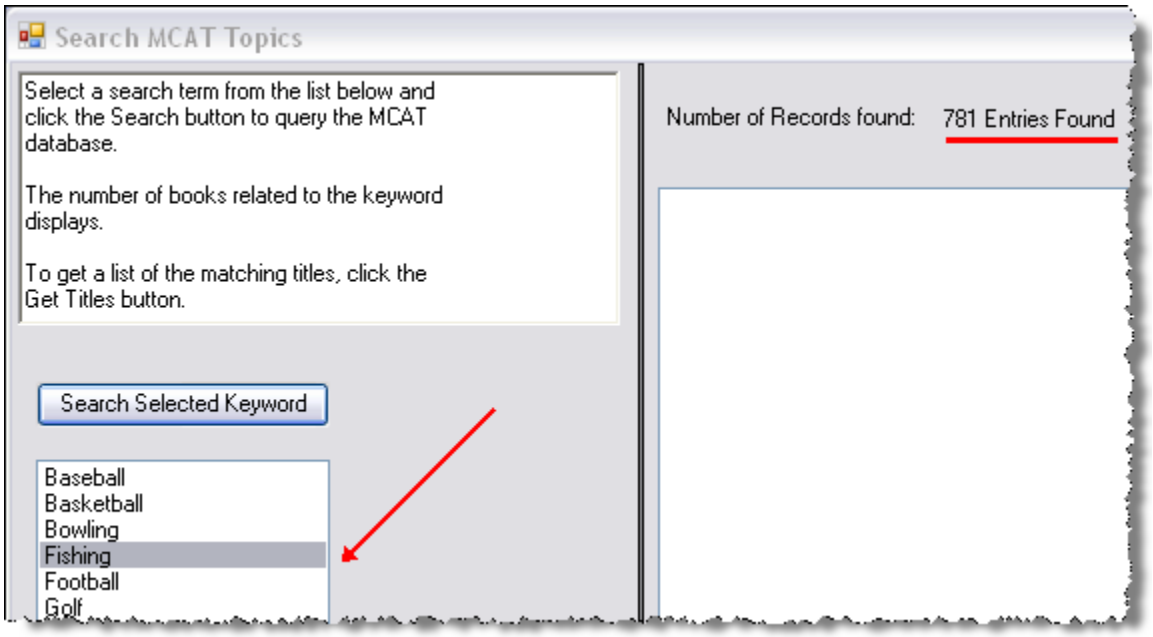
4. Select a keyword from the listbox and click the Search Selected Keyword button. The following project execution sequence should occur:

a) The text "k=*your keyword* selection" displays in the Next Command field.

b) Search is initiated in the host application.

c) The UMich Online Catalog Keyword Index (Search Results) screen displays showing the number of records found in the Search Results field and a listing of records (Date, Title, Author) in rows on the screen.

d) The number of records found displays on the Windows Form next to the "Number of Records Found" label. Examples of the session window and Windows Form follow:

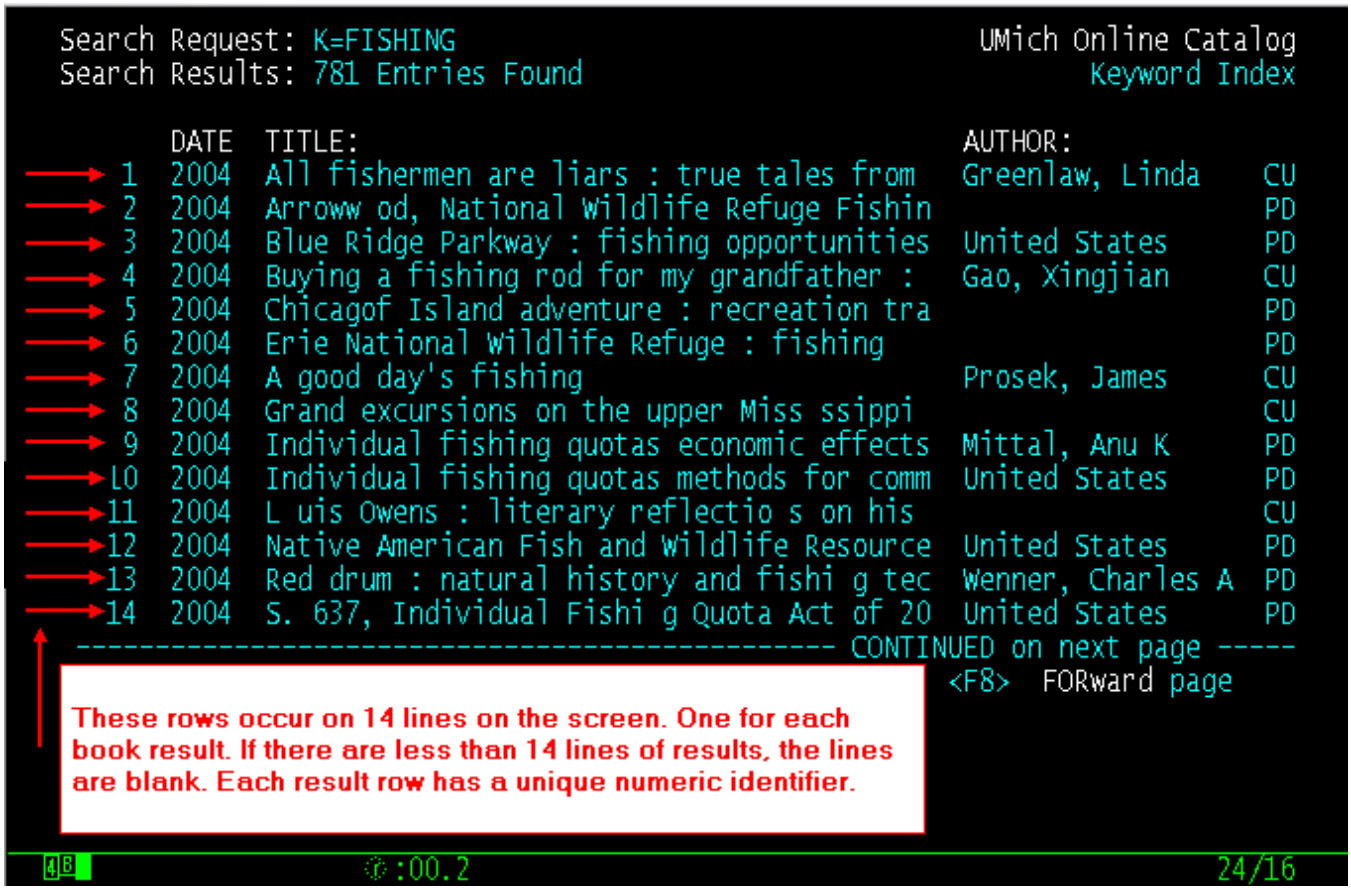


5. Click the Stop Debugging button in OpenSpan Studio to unload the project and stop the OpenSpan Runtime application.

This page left intentionally blank.

CHAPTER 3: USING PARTIAL SCREENS

In host/mainframe applications, the screen may repeat similar types of information in multiple locations. For example, using the solution created in this training, rows of titles and authors appear which represent books matching the query term (the example below shows the results for k=fishing):



If you want to create an automation that retrieves the text from each row of results shown in a screen like the sample above, you have the following options:

- Interrogate each line as a Region. This results in 14 Region controls per screen.
- Interrogate the line as a Partial screen. You can define a partial screen as an area within an already interrogated parent screen. The partial screen has the option of appearing multiple times and in multiple places within the parent screen. This results in a single Partial screen control required to return all of the rows of the results.

This chapter describes how to create and use partial screens in OpenSpan text adapter projects. The solution used in the previous chapters is continued to add logic for returning the Titles of records matching the search keyword in the MCAT database.

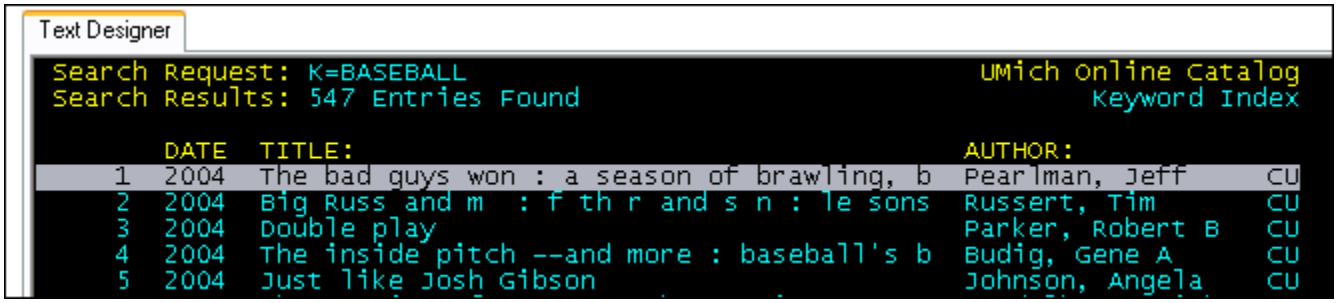
Interrogate the Results Screen Using Partial Screen Matching

1. In OpenSpan Studio, start the emulator application by selecting the Windows adapter project item (WinEulator.os) and clicking the Start Interrogation button in the Designer. The emulator application launches and the UMICH.edp session loads.
2. Open the **Text Adapter** project item, GenWinHllapi.os, and select **Start Interrogation** from the **Designer**. **University of Michigan Libraries Database Selection Menu** screen is loaded into the OpenSpan Designer. Interrogate the session using the screen displayed in the Designer for the text adapter.

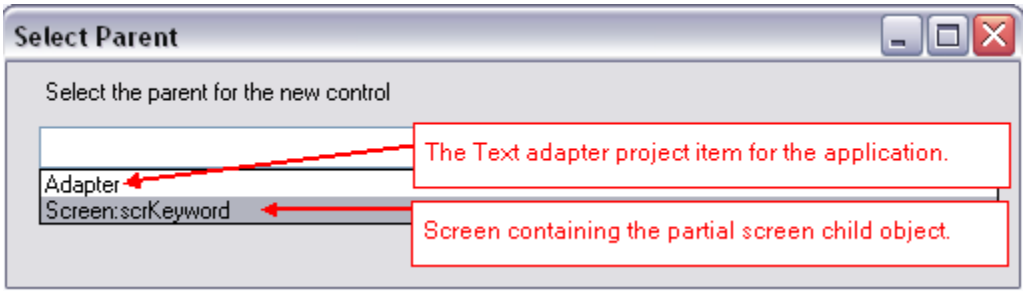
Type MCAT in the Database Selection field and press **Enter** to navigate to the UMich Online Catalog Introduction screen.

3. Type k=baseball in the Next Command field and press Enter to perform a keyword search. The UMich Online Catalog Keyword Index (Search Results) screen displays. The screen lists 14 titles which match the search criteria.

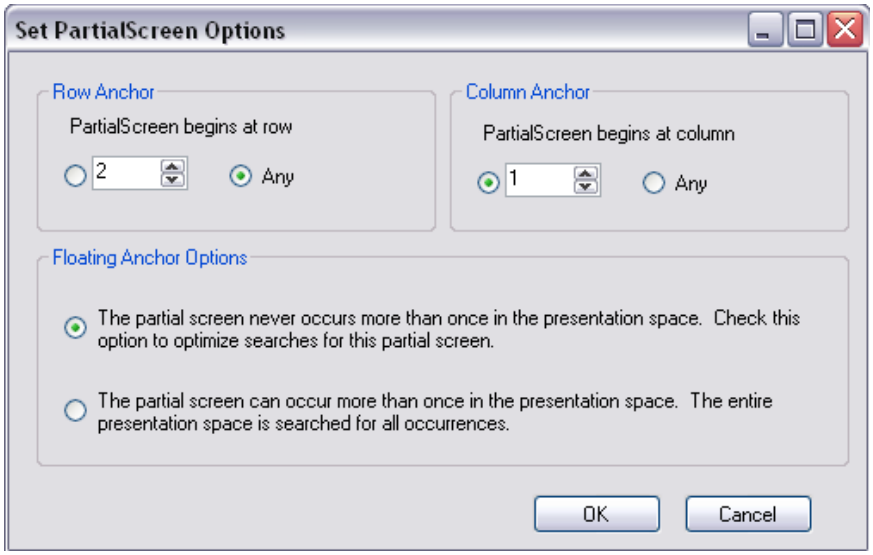
Highlight the first row of results as shown in the following illustration:



4. Right-click on the highlighted row to view the interrogation options.
5. Select **Save as Partial Screen**. The Select Parent dialog displays.
6. Use the Select Parent dialog to identify the main screen which must be matched before the partial screen is matched. If the partial screen can appear on any screen within the application, use the Emulator as the parent. An example of the Select Parent dialog follows:



7. For this training module, select the Screen:scrKeyword as the parent and click **OK**. The Select Parent dialog closes and the Set PartialScreen Options dialog opens. An example of this dialog follows:



8. The result row always starts in Column 1 and can appear at any row position within the screen. Therefore, set the partial screen options as follows:

   - Row Anchor – Any
   - Column Anchor – 1
   - Floating Anchor Options – "The partial screen can occur more than once in the presentation space. The entire presentation space is searched for all occurrences."

9. Click **OK** to save these settings and dismiss the Set PartialScreen Options dialog.

Partial Screen Match Rule Using Complex Match Rule Option

In this section you will set a matching condition for the partial screen. The matching condition specifies text that appears in all occurrences of the partial screen. In this case, the text that appears for each partial screen is a record number (1, 2, …).

1. Highlight the result number in the partial screen (1) and the two spaces preceding the number – this allows for record numbers up to 3 digits. The easiest way to highlight this area is to left-click immediately following the number 1 and then use the left arrow key to highlight the number and two preceding spaces. See the following illustration:

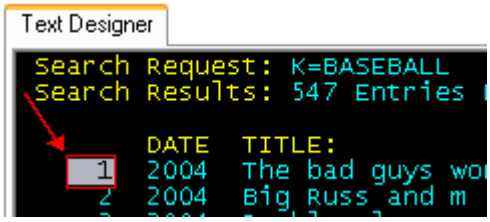
Use the Set PartialScreen Options dialog to define where the partial screen appears within the application screen(s). A partial screen location is defined by row and column.

For example, if the partial screen always appears at a certain row on the screen, you can set the row number. Likewise, if the partial screen always appears at a certain column, you can set the column number.

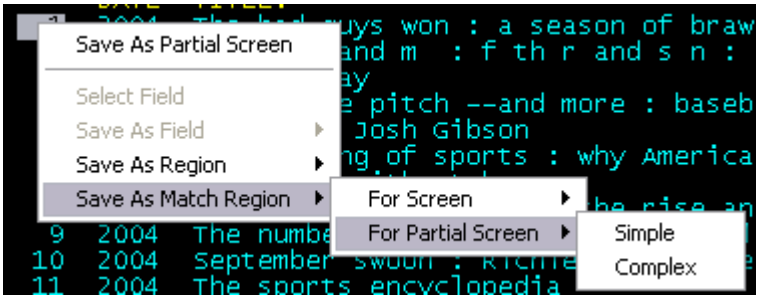
Depending on how the partial screen appears within the parent screen, you have the options to set:

- Specific Row – Any Column (Floating horizontally)
- Specific Column – Any Row (Floating vertically)
- Specific Row and Specific Column
- Any Row and Any Column

If the partial screen floats vertically or horizontally (starts at any row and/or any column), you must set the Floating Anchor Options to define whether the partial screen appears multiple times within the parent.



2. Right-click on the highlighted area and select **Save As Match Region | For Partial Screen | Complex:**



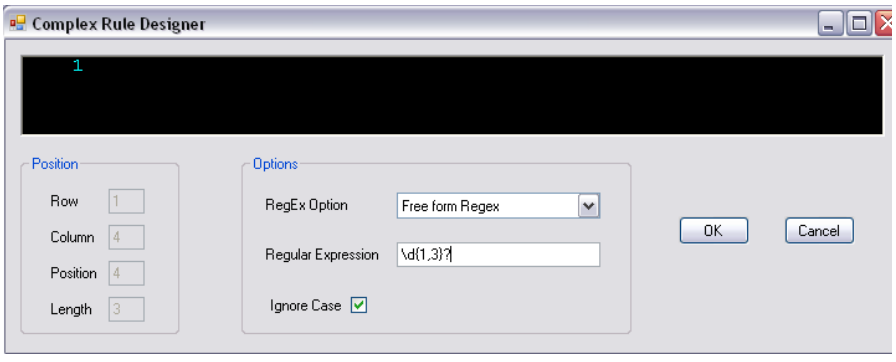
3. Since the record number differs for each record, the Complex matching option is required. When you select this option, the Complex Rule Designer displays.

4. Use the Options groupbox to set the following matching requirements:

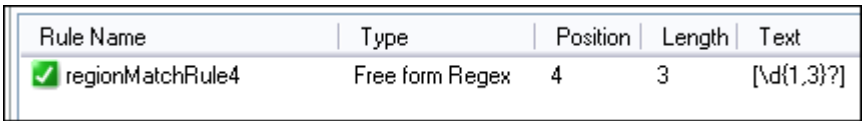
   a. RegEx Option – Free form Regex

   b. Regular Expression - \d{1,3}? (match any three digits followed by any character)

   c. Leave the Ignore Case option as default. Your Complex Rule Designer dialog should look like the following:



5. Click **OK** to apply the match rule and dismiss the Complex Rule Designer dialog. The Match Rules for Partial Screen displayed on the Designer should look like the following:



Use the Complex Rule Designer to set Position and text matching Options which OpenSpan uses to match the selected Match Region text.

For a partial screen, the position options are already set through the Set PartialScreen Options dialog.

Use the Complex Rule Designer dialog to set the way OpenSpan matches the selected text.

In the Options groupbox, set the RegEx option to define the type of regex syntax to use.

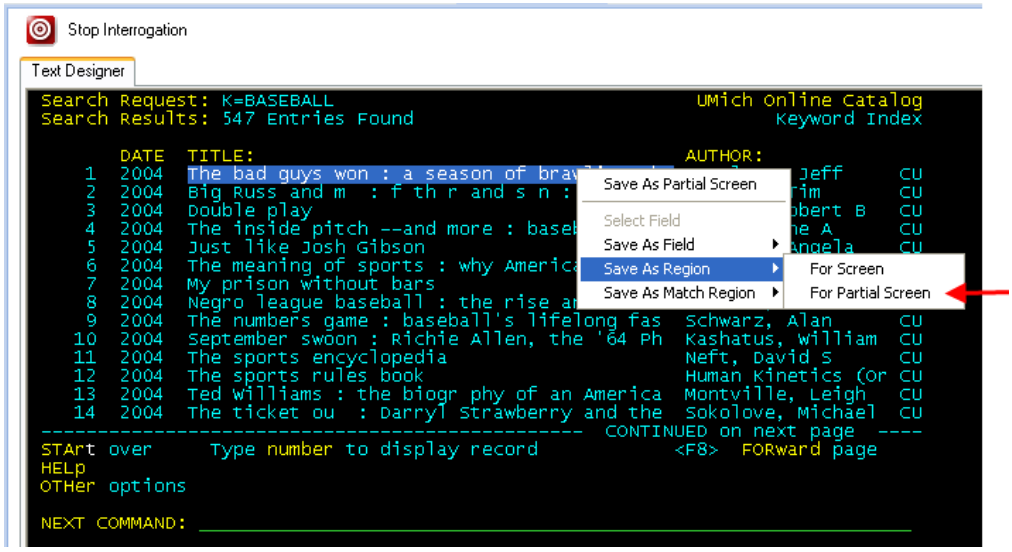
In the Selected Text option, set the text to match for the Contains or Does not contain RegEx options.

If you select Free Form Regex for the Regex Option, the Selected Text option changes to Regular Expression for you to enter the regex syntax to match the text.

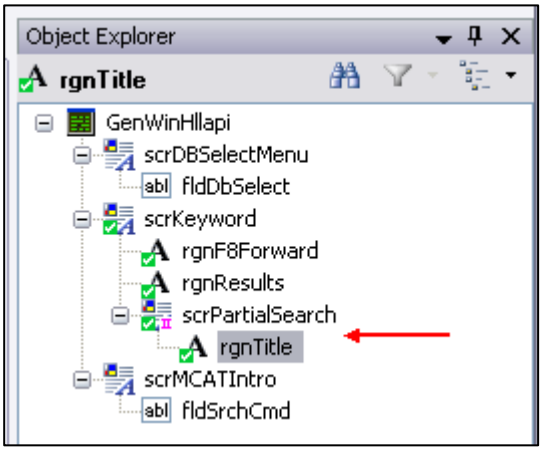
6. Highlight the Partial Screen in the Object Explorer and use the Properties window to rename the control: **scrPartialSearch**.

Interrogate Title Text

1. Highlight the Title for the record in the partial screen, right-click and select **Save As Region | For Partial Screen**. A region control is added to the Object Explorer under the scrPartialSearch screen.



2. Highlight the region in the Object Explorer and use the Properties window to rename the control: **rgnTitle**. Your Object Explorer should look like the following:



3. Click **Stop Interrogation** for the GenWinHllapi.os project item.
4. Open the WinEmulator.os project item in the Designer and click the **Stop Interrogation** button.
5. Save all solution files.

Add Automation to Get Titles for Matching Records

This automation will loop through the results listed after the user has searched on a keyword in the MCAT database and records have been found. When the user clicks the Get Titles button on the Windows Form, the automation reads each Title in the rgnTitle control (interrogated in the previous section) and adds the title text to a listbox on the Windows Form. The automation will prompt the user as follows:

- If 15 or more titles are returned, the user is prompted: "Retrieve more titles?". If the user chooses Yes, then the next screen of titles is read and added to the listbox. If the user chooses No, then the adapters are shut down and search disabled.

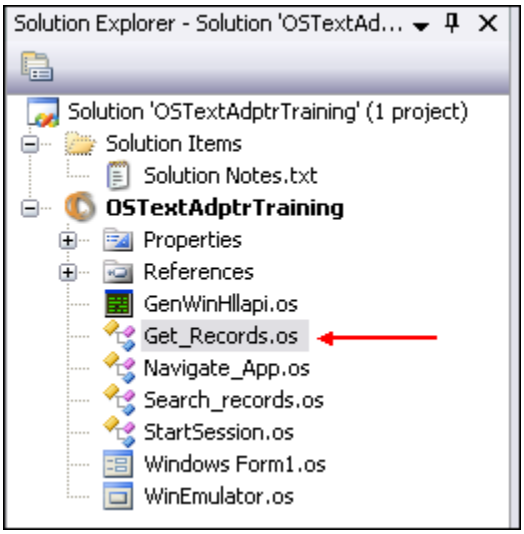
- If there are no more titles available or if the search returns less than 14 titles, the user is prompted: "No more records. New Search?". If the user chooses yes, the Windows Form results display is rest and a new search can be initiated. If the user chooses No, then the adapters are shut down and the search disabled.

This automation contains four main sections:

- Processing result rows and yielding title text on the first screen of results.
- Halting the row processing when there are less than 14 titles listed on a screen
- Processing the next screen of results if more than 14 titles are returned.
- Ending the retrieval process.

Begin by highlighting the OSTextAdptrTraining project in the Solution Explorer.

1. Right-click on the project and select **Add | New Automation** from the local menu. The Automation is added to the project and displays in the Solution Explorer and Object Explorer.
2. Select the Name property (in the Properties window) for the automation and rename it: **Get_Records.os**. Your Solution Explorer should look similar to the following:



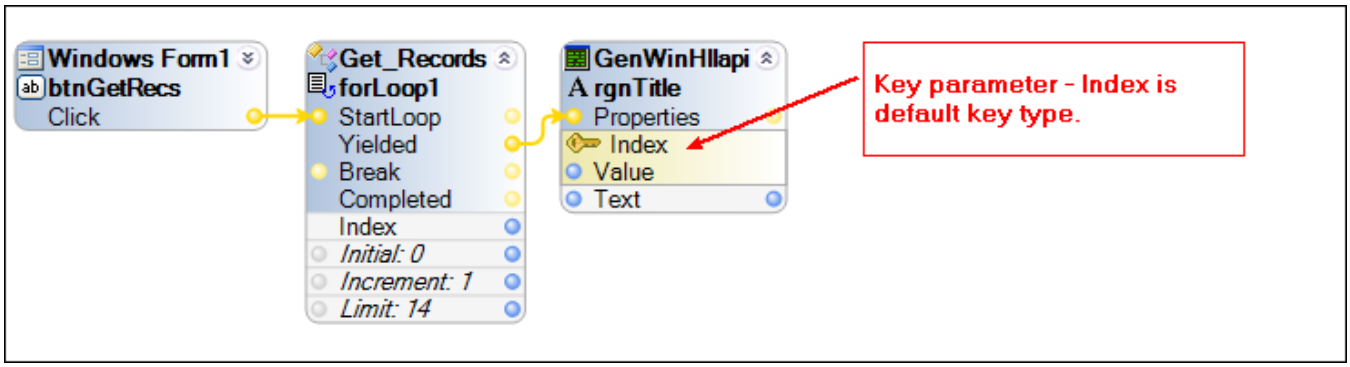
3. Add two OpenSpan messageDialog components to the automation – Local tab. Select the "(String message); DialogResult overload option for both dialogs. Name the messageDialog components: **msgNoMoreRe**c and **msgGetMore**.

4. Set the properties for the messageDialogs as follows:
   - msgNoMoreRec – "**No more records. New Search?**". This message will display if OpenSpan detects there are no more titles to retrieve. Set the Buttons property to: YesNo.
   - msgGetMoreRec – "**Retrieve more titles?**". This message will display if OpenSpan detects that an additional screen of titles is available.
5. Add a forLoop component to the automation. Set the Limit parameter on the connection block to 14 (the maximum number of results returned by the search per screen).
6. Add two Jump labels to the automation. Rename the Jump Labels: **Break** and **EndRetrv**. These Jump controls will be used to route the event path in the automation.
7. Add a stringUtils component to the automation. Use the Object Explorer to select the Contains method for the component. Set the seekString parameter to **F8**. This method will be used to determine whether additional screens of search results are available.
8. Add the following properties, methods and events to the automation:
   - WindowsForm1.btnGetRecs.Click event – this event will initiate executing the automation logic.
   - WindowsForm1.lbxTitles.Items.Add(1 param obj) method – this method adds each title to the listbox on the Windows Form.
   - WindowsForm1.btnGetRecs.PerformClick() method – this event will re-initiate reading the result records.
   - GenWinHllapi.rgnTitle.Text property – populates the listbox on the Windows Form.
   - GenWinHllapi.rgnF8Forward.Text property – used to verify addition screen of results exists.
   - GenWinHllapi.rgnF8Forward.Clear() method – deletes previous keys/commands sent to the field.
9. Save all solution files.

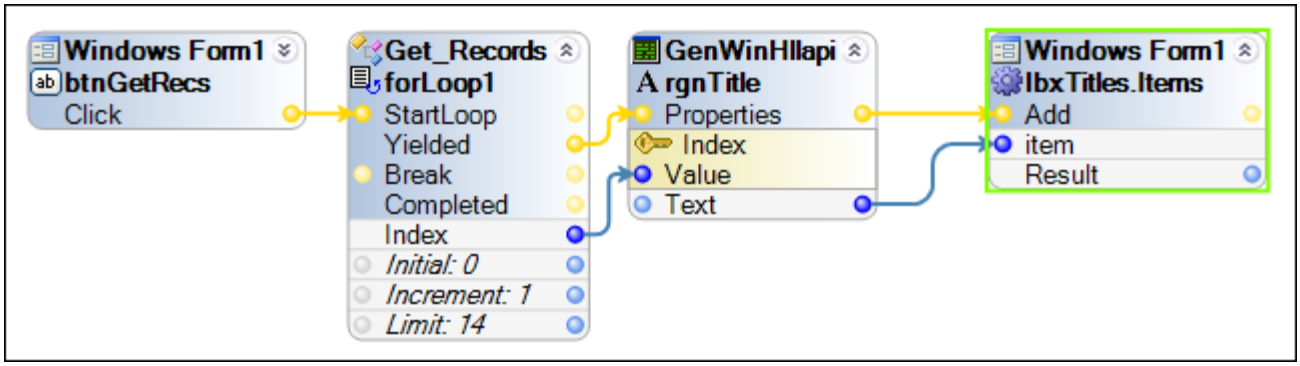
Establish Logic to Read Screen Rows and Yielding Title Text for First Screen of Results

1. Use the OpenSpan forLoop component to read each line of result text and return the Title (rgnTitle.Text property). Arrange the connection blocks and connect the event path as shown in the following image:

   Note: When you connect the event path for the rgnTitle control, a Key parameter displays. This is because multiple instances of this control can occur (clones). You must specify which instance to use. The options are: Last, Index, None, First, and Current. For this training, the default key type "Index " applies since it can be linked to the record number/forLoop Index.



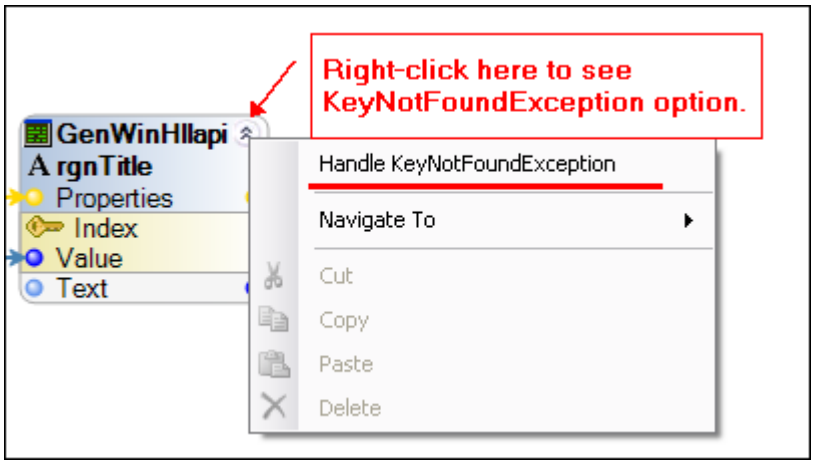
2. Connect the Index output from the ForLoop to the Value for the Index Key of the rgnTitle.Text property. Complete the connections as follows:



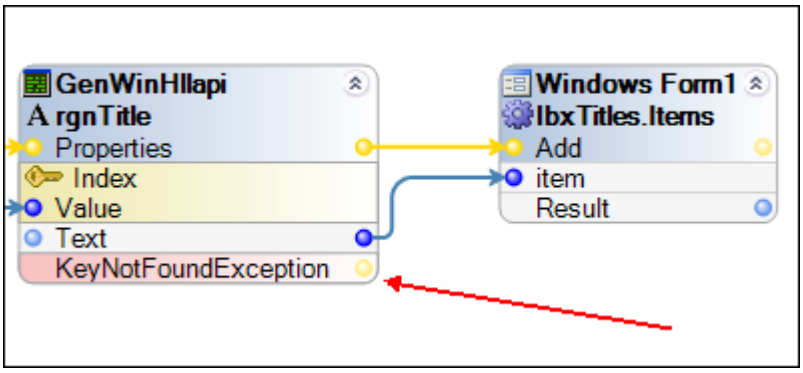
Establish Logic to Halt Processing when Less than 14 Titles Listed on Screen

Add logic to handle the case where the Index of the forLoop does not correspond to the Index of any of the rgnTitle controls. For example, suppose only 5 results are returned for the search. The forLoop would cause an exception error when the loop index is 6 and the automation attempts to get the Text for the rgnTitle having an index of 6 since this control does not exist.

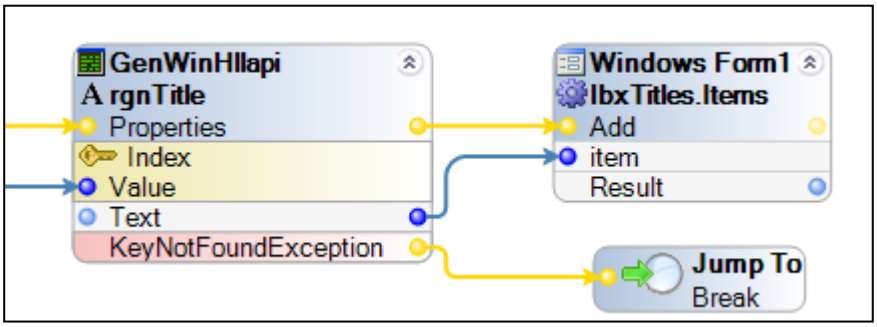
1. Right-click on the top of the rgnTitle.Text property connection block to expose the Handle KeyNotFoundException option as follows:



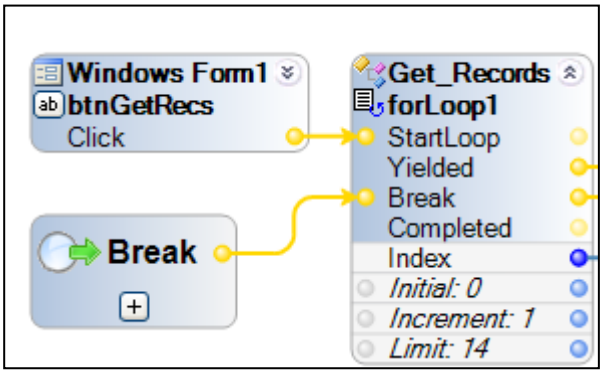
2. Select the Handle KeyNotFoundException option. A new output event is added to the connection block for the control:



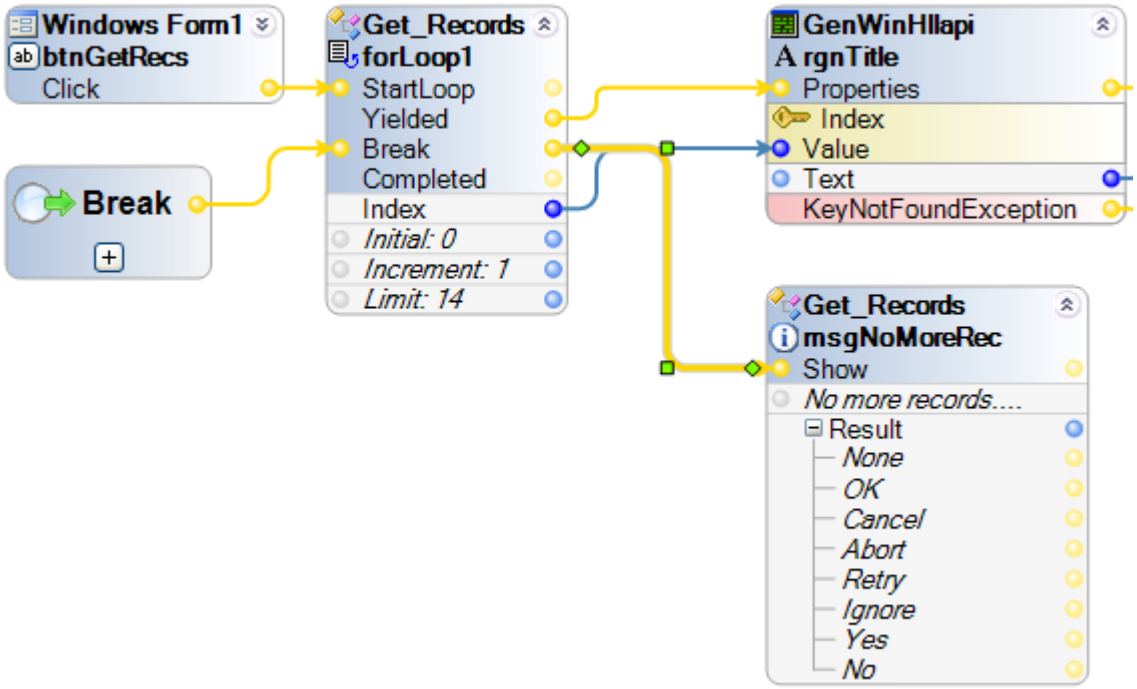
3. When the index of the loop is greater than the number of results returned, the forLoop should stop processing. To stop the loop, trigger the Break input event. Use the Break Jump label to trigger the Break event. Right-click on the automation and select **Jump to | Break**. A Jump To label is added. Connect the KeyNotFoundException event to the Break Jump To label.



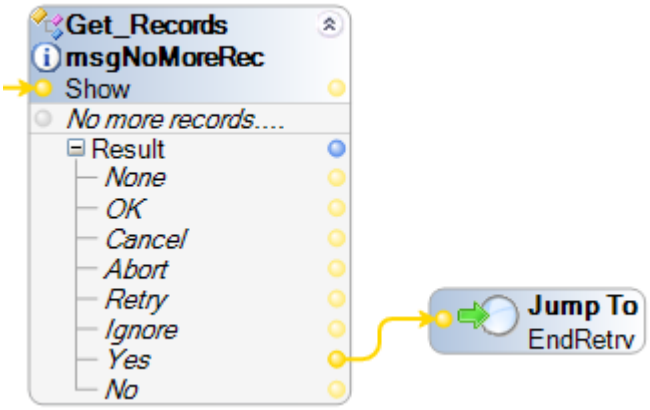
4. Connect the Break Jump label to the Break input event for the ForLoop:



5. When the forLoop receives the Break event input, the loop stops processing and the Break output event is triggered. When the Break output event is raised, use the msgNoMoreRec message to inform the user that there are no more titles to return for this search. Connect the Break output event from the forLoop to the Show input event of the msgNoMoreRec messageDialog component:



6. The msgNoMoreRec message displays the following: No more records. New Search? The options are Yes and No. If the user selects Yes – New Search, then the automation executes the logic to reset the host application to the initial search input screen (UMich Online Catalog Introduction (Welcome to MCAT). Right-click on the automation and select **Jump to | EndRetrv**. A Jump To label is added. Connect the Yes output event from the msgNoMoreRec to the EndRetrv Jump To label:



Note: If the users selects No for the "No more records. New search" message, the Text and Windows adapters will be stopped. The automation for this process will be added later in this chapter.

7. Save all solution files.

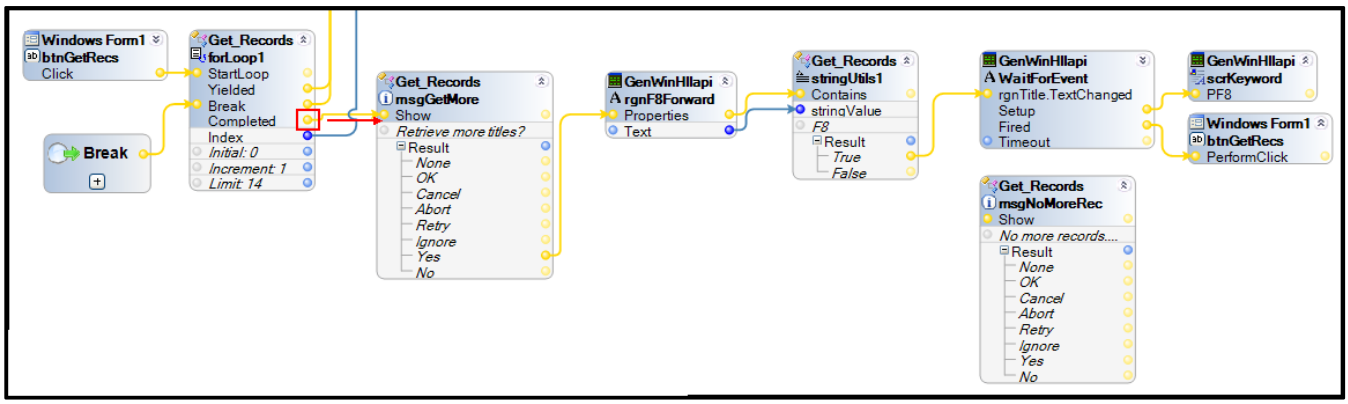
Establish Logic to Process Additional Results Screens

The host application retrieves all of the titles that match the search criteria and displays up to 14 results on the screen. If more than 14 results are found, the text <F8> FORward displays at the bottom right of the screen (you interrogated this text – the rgnF8Forward region). On the last screen of results, the text in this area changes to: <F7> BACK page.

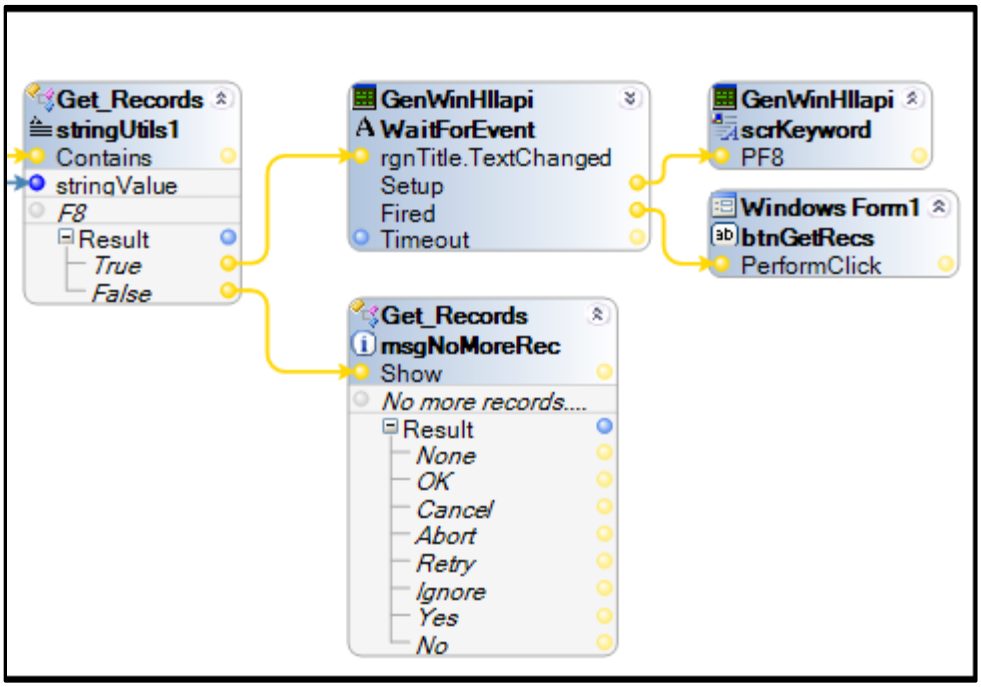
When the rgnF8Forward text no longer contains the string "F8" then the application is displaying the last screen of results. Using the forLoop.Completed event and the text property from the rgnF8Forward, you can add logic that enables the user to continue retrieving titles and then stop processing on the last screen of search results. Once the forLoop processes 14 rows and adds the titles to the Window Form lbx.Titles listbox, the loop raises the Completed event. At this point, the user can be given the option to retrieve more titles or end the processing.

In this part of the automation, add logic to prompt the user to continue retrieving titles and re-start the forLoop to retrieve the data. If the user chooses to retrieve more data but there are no more screens of results data (the rgnF8Forward text does not contain the string F8), a message displays indicating that there are no more records.

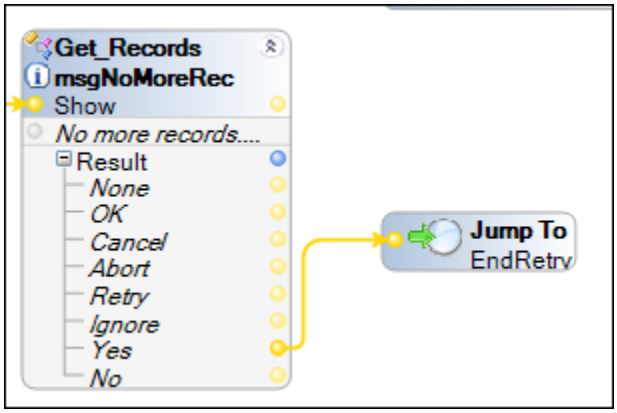
1. Begin by connecting the Completed output event from the forLoop to the Show method for the msgGetMore messageDialog. This will cause the message "Retrieve more titles?" to display once the first 14 results have been retrieved.
2. If the user selects Yes, the automation must check to see if an additional result screen exists. If another results screen exists, the automation must navigate to the result screen and start the forLoop again to retrieve the data from the screen. Add the following to the automation:
   - GenWinHllapi.rgnTitle.TextChanged event – after adding this event, right-click on the event and select "Wait For This Event". This will be used to raise the TextChanged event after the PF8 (Forward) command occurs. When the event is raised, the PerformClick method is called on the Windows Form GetRecs button which starts the forLoop again.
   - GenWinHllapi.scrKeyword.PF8() method – sends PF8 command to screen.
3. Arrange the connection blocks and connect the event path as shown in the following image:



4. If the rgnF8Forward region does not contain the string F8, the host application is on the last screen of results. In this case, the StringUtils.Contains method yields a FALSE result and the False event is raised. Connect this event to the msgNoMoreRec messageDialog Show method. This informs the user that there are no more records to retrieve.

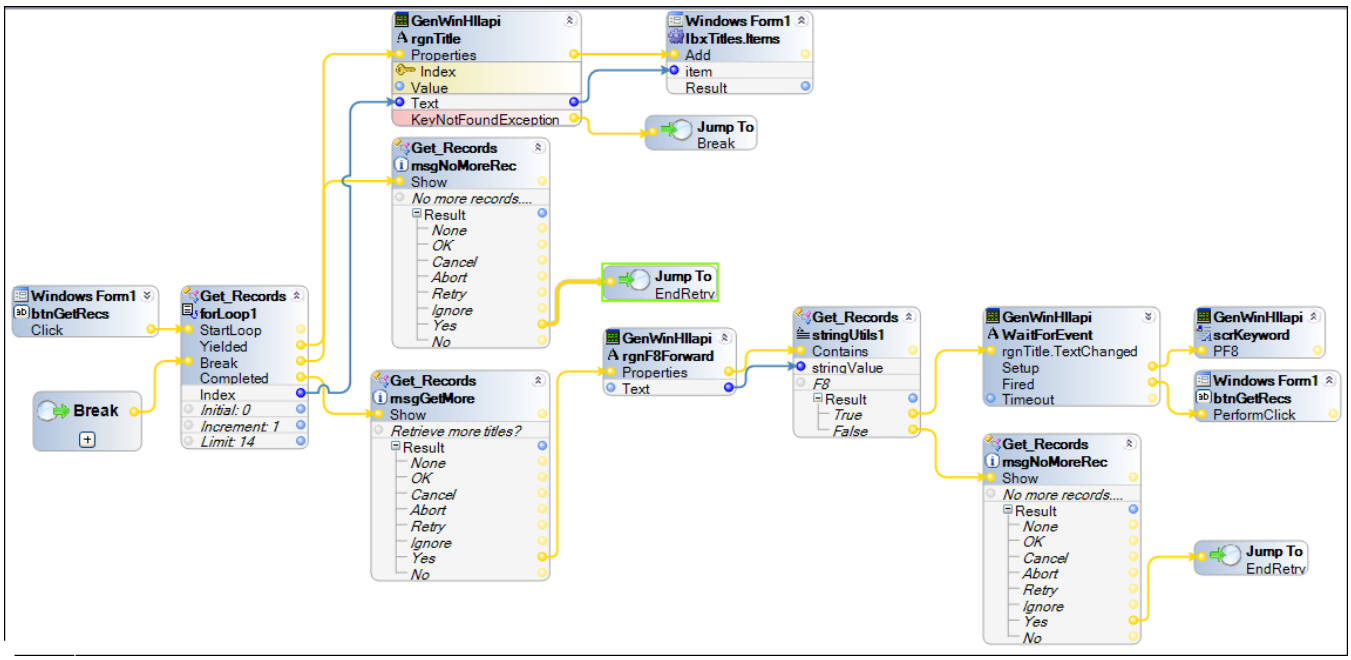


5. The msgNoMoreRec messageDialog displays the text: No more records. New Search? The options are Yes and No. If the user selects Yes – New Search, then the automation executes the logic to reset the host application to the initial search input screen (UMich Online Catalog Introduction (Welcome to MCAT). Right-click on the automation and select **Jump to | EndRetrv**. A Jump To label is added. Connect the Yes output event from the msgNoMoreRec messageDialog to the EndRetrv Jump To label.



6. Save all solution files.

Your automation should look like the following:



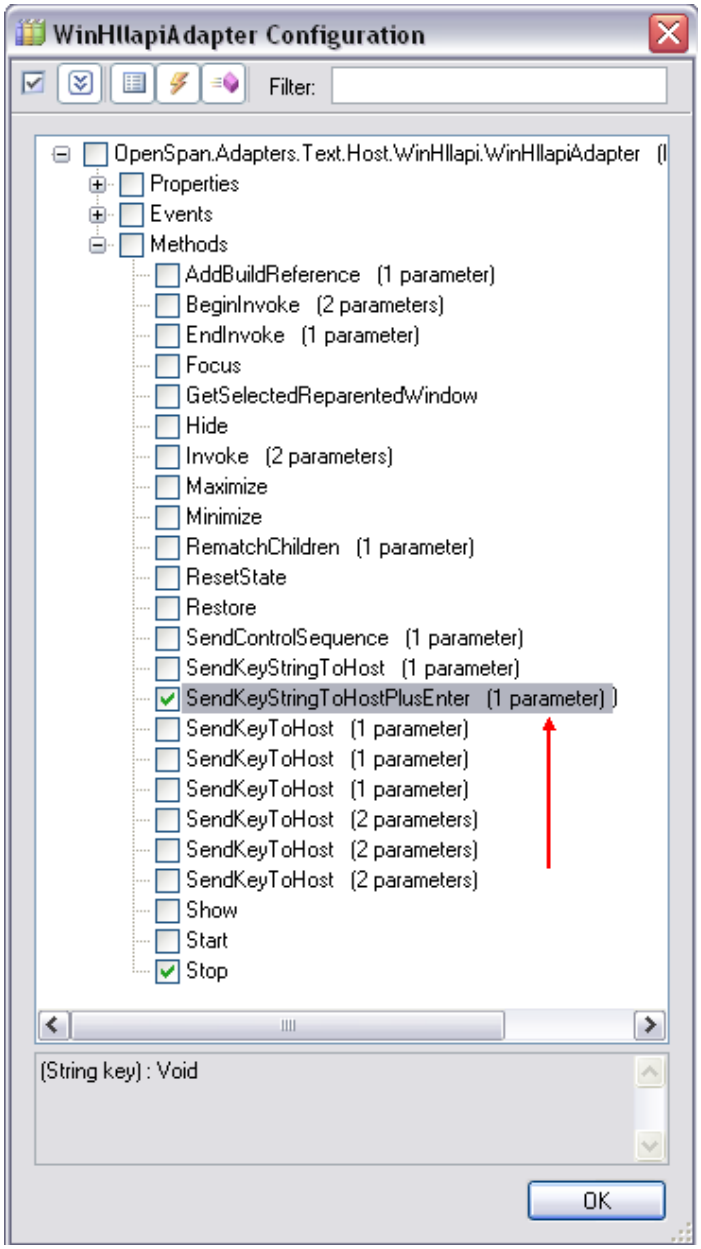
Establish Logic to End the Retrieval Process

When the automation has retrieved all available records for the search, the user is notified via the msgNoMoreRec messageDialog: No more records. New Search?

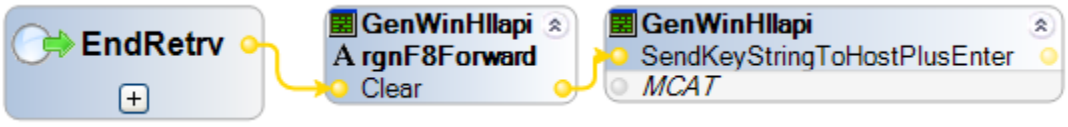
If the user selects Yes to conduct a new search, the automation must navigate the host application back to the UMich Online Catalog Introduction (Welcome to MCAT) screen so that new search criteria can be entered. For this application, the command “MCAT” returns the application to the MCAT Introduction screen. The Text adapter method SendKeyStringToHostPlusEnter can be used to navigate the application to the MCAT Intro screen.

Begin by adding the SendKeyStringToHostPlusEnter() to the automation:

1. Highlight the GenWinHllapi adapter in the Object Explorer. Using the Object Inspector, select the SendKeyStringToHostPlusEnter(String key).



2. Arrange the connection blocks and connect the event path as shown in the following image:

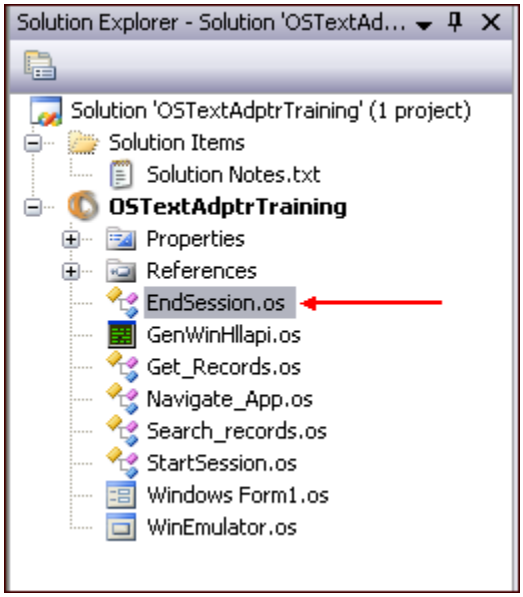


3. Save all solution files.

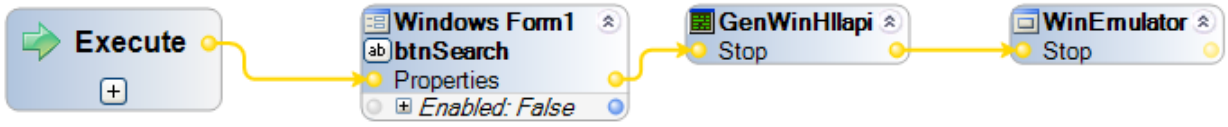
Add Automation to Stop Adapters

The user has the option to stop searching when the msgNoMoreRec message “No more records. New Search?” displays or when the msgGetMore message “Retrieve more titles?” displays. In either case, if the user selects No, then the automation should disable the search button on the Windows Form and shut down the adapters. Create an automation to end the search processing as follows:

1. Add a new automation to the project. Name the automation: **EndSession.os**. The new automation displays in the Solution Explorer.

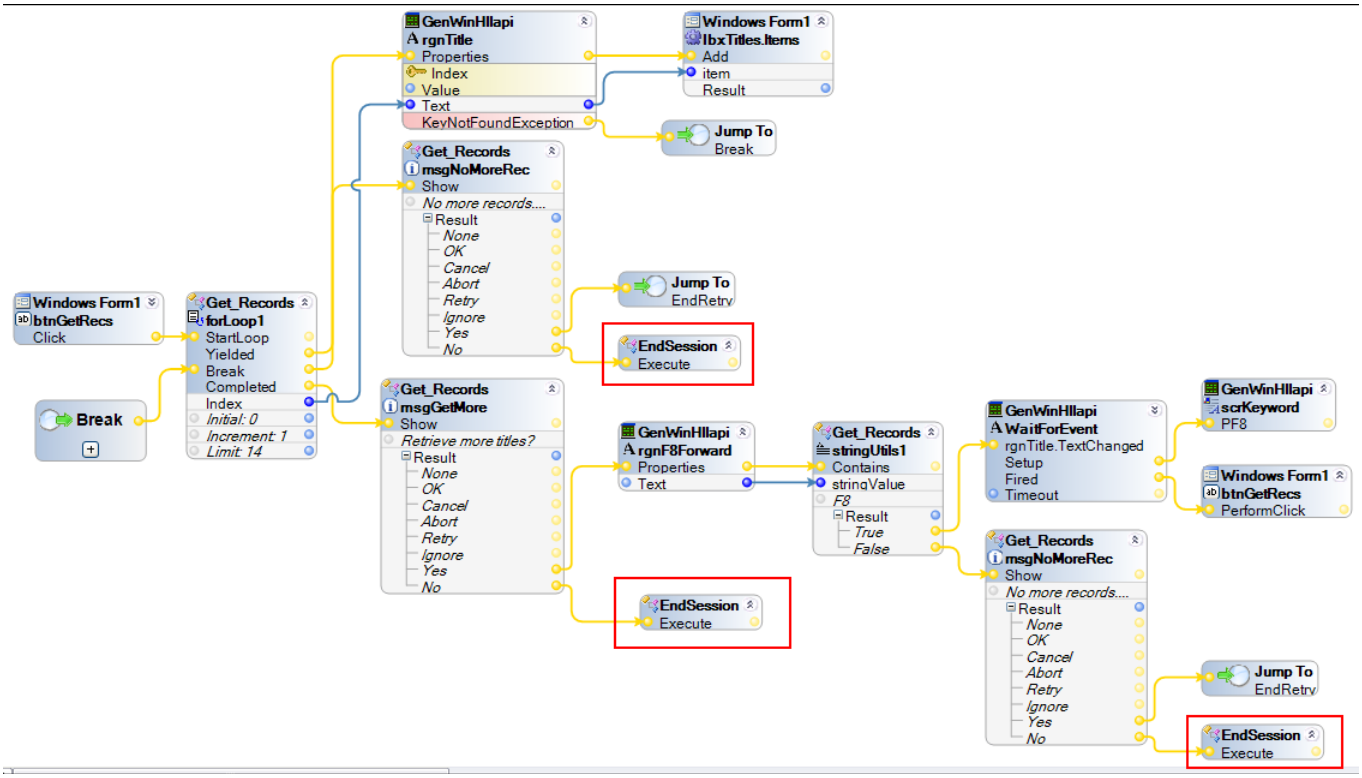


2. Right-click on the automation and add an Entry Point. An Execute block appears on the automation.
3. Add the following properties, methods and events to the automation:
   - WindowsForm1.btnSearch.Enabled property – set the Enabled property to False on the connection block.
   - GenWinHllapi.Stop() method – this method stops the text adapter which ends the session.
   - WinEmulator.Stop() method – this method stops the Windows adapter which closes the emulator application.
4. Arrange the connection blocks and connect the event path as shown in the following image:



5. Save all solution files.

6. Return to the Get_Records automation and add the EndSession.Execute() **method** as follows:



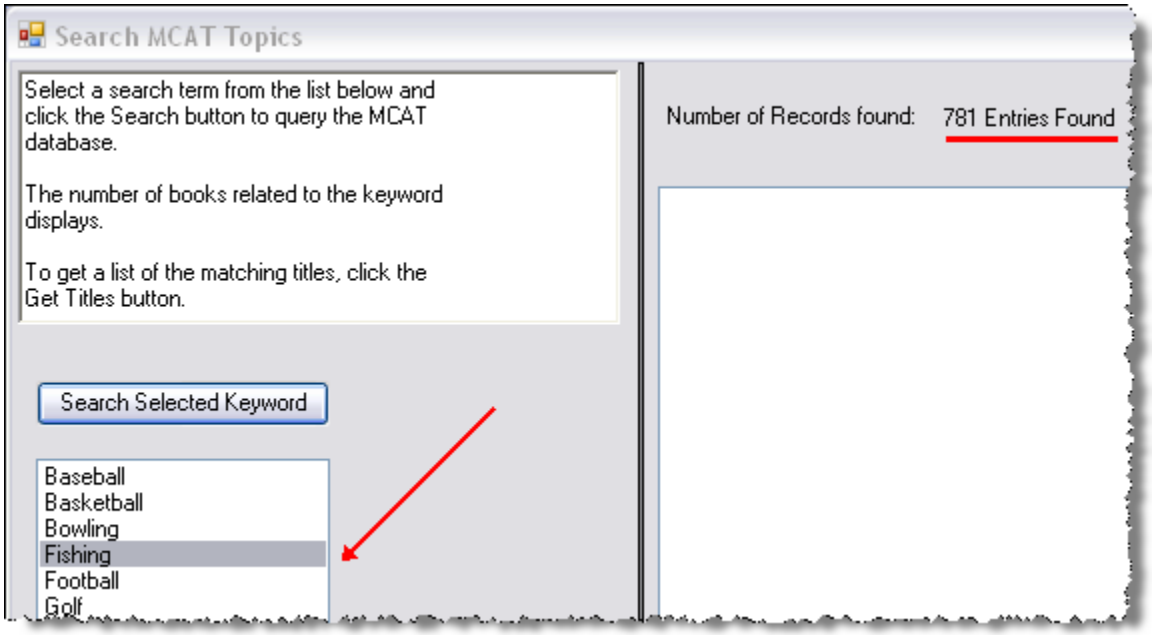
7. Save all solution files.

Build and Debug the Solution

Make sure that the UMICH_Large.bat file is running and then do the following:

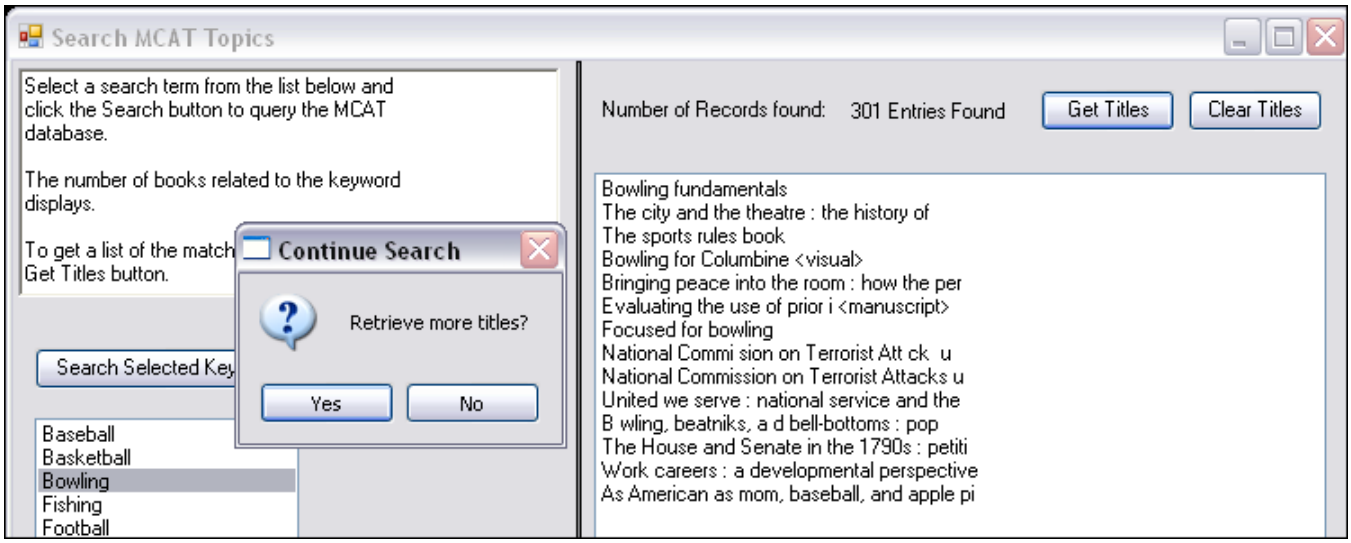
1. Select the Debug solution configuration from the Debug toolbar.
2. Click the **Debug** button. OpenSpan Compiles the project and launches a design version of the project in the OpenSpan Runtime application installed with OpenSpan Studio (or Plug-in).
3. The following project execution sequence should occur:
    a) The project is built and no build errors are reported.
    b) OpenSpan Runtime loads the design project OSTextAdptrTraining.
    c) OpenSpan Runtime loads the Search MCAT Topics windows form. Note that the search term list box is disabled. This listbox only becomes enabled when the MCAT query screen is displayed in the emulator.
    d) OpenSpan Runtime launches the emulator (myEXTRA! Enterprise) and opens the UMICH.edp session.
    e) The initial Database Selection Menu screen displays in the emulator.
    f) The emulator navigates the host application to the UMich Online Catalog Introduction (Welcome to MCAT) screen.
    g) The keyword listbox on the Windows form becomes enabled.
4. Select a keyword from the listbox and click the **Search Selected** Keyword button. The following project execution sequence should occur:
    a) The text “k=*your keyword* selection” displays in the Next Command field.
    b) Search is initiated in host application.
    c) The UMich Online Catalog Keyword Index (Search Results) screen displays showing the number of records found in the Search Results field and a listing of records (Date, Title, Author) in rows on the screen.

d) The number of records found displays on the Windows Form next to the “Number of Records Found” label. Examples of the session window and Windows Form follow:

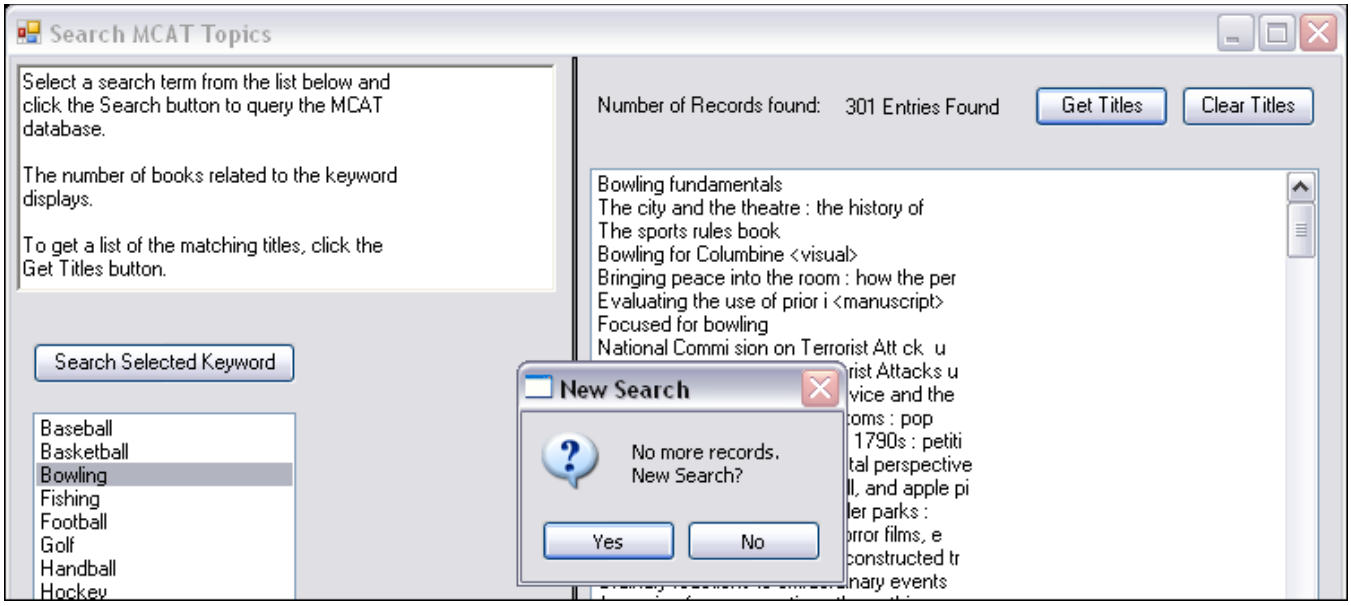


5. Click the **Get Titles** button in the Windows Form. The list of titles from the host application displays in the listbox on the Windows Form.

- If another screen of results is available, the message “Retrieve more titles?” displays.
- If there are no more results, the message “No More Records. New Search?” displays.



6. Click **Yes** for the “Retrieve more titles?” message until you receive the “No More Records” message.



7. Click **Yes** for the “No More Records. New Search?” message and then click the **Clear Titles** button on the Windows Form.
8. Select **Windsurfing** from the Keyword listbox and click **Search Selected Keyword**. The number of search results displays: 8 Entries.
9. Click the **Get Titles** button. Titles populate the listbox and the message “No More Records. New Search?” displays.

10. Click **No**. The emulator closes.
11. Click the **Stop Debugging** button in OpenSpan Studio to unload the project and stop the OpenSpan Runtime application.

SUMMARY

This training module detailed the following:

- Configuring the Windows adapter to integrate a terminal emulator application
- Configuring and using the Generic WinHllapi adapter to integrate screens and fields from a mainframe application
- Interrogating screens, fields, and regions within a mainframe application
- Creating automations to navigate a mainframe application and return data from the application

This page left intentionally blank.

APPENDIX A: USING THE TRACEPLAYER APPLICATION

A trace of a recorded host session rather than a live host session will be used for this exercise.

To start the recorded host session for the University of Michigan Libraries system, run the UMICH_Large.bat file. If you are using a VM provided by OpenSpan for this course, the .bat file is located in the C:\Traceplayer-1 folder. An example of the traceplayer output is shown below: