



OpenSpan Elective Training

Using the OpenSpan Siebel Adapter

- **CHAPTER 1:** Using OpenSpan Studio with Siebel
- **CHAPTER 2:** Sample Project – Data Transfer
- **CHAPTER 3:** Navigation Controls
- **CHAPTER 4:** Working with Pop-up Dialog Boxes

© Copyright 2010 OpenSpan, Inc. All Rights Reserved

No part of this publication may be reproduced or distributed in any form or by any means, electronic or otherwise, now known or hereafter developed, including, but not limited to, the Internet, without explicit prior written consent from OpenSpan, Inc. Requests for permission to reproduce or distribute to individuals not employed by OpenSpan any part of, or all of, this publication should be mailed to:

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

OpenSpan[®] is a registered trademark of OpenSpan Inc., a Georgia Corporation.

Siebel is a trademark or registered trademark of Oracle Corporation and/or its affiliates.


Microsoft[®], Visual Studio[®], MSDN[®], and Windows[®] are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

CONTENTS

- Chapter 1: Using OpenSpan Studio with Siebel5**
 - Prerequisites 5
 - Supported..... 5
 - Solutions Used in this Training Module 5
 - Setting up Internet Explorer to Use Siebel..... 5
 - Identifying Web Application Pages and Controls..... 6
 - Siebel Application Pages 6
 - Siebel Application Controls - ElementID Match Rule..... 7
- Chapter 2: Sample Project – Data Transfer9**
 - Automating the Login Process 9
 - Get Data from Siebel List 11
 - Initiate a Process 15
- Chapter 3: Navigation Controls..... 19**
 - Exercise – Automating Navigation 21
 - Create Navigation Controls..... 21
 - Complete Automate Navigation Solution 25
- Chapter 4: Working with Pop-up Dialog Boxes 27**
 - Exercise – Popup Dialog 27

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 that require emphasis, such as command buttons, menus, 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.
<u>Remember</u>	<u>Definitions of terms and important concepts that bear remembering.</u>
	Next to the Tip icon, you can find best practices and shortcuts to use OpenSpan Studio more effectively.

CONTENTS

This page intentionally left blank.

Chapter 1: Using OpenSpan Studio with Siebel

OpenSpan projects can be used to integrate with Siebel and automate Siebel user actions. This document provides details on integrating Siebel with OpenSpan projects and includes step-by-step instructions for creating sample OpenSpan Siebel projects.

Prerequisites

This guide requires a working knowledge of OpenSpan Studio. The tutorial also assumes that you are familiar with the Siebel application.

The course exercises require your system to be setup with the following:

- OpenSpan Studio 4.5
-

Supported

OpenSpan Studio supports Siebel versions 7.5, 7.7, 7.8 and 8.0. The OpenSpan web adapter is used to integrate Siebel into OpenSpan projects. Since Navigation elements do not exist in Siebel 7.5, OpenSpan Studio cannot interrogate them. Furthermore, the methods GetHeading, GetHeadings, and GetColumnFromHeading are not implemented for Siebel 7.5. For more information on Siebel properties, methods, and events see the OpenSpan online Help topic: **Siebel Control – Properties, Methods, and Events**.

Solutions Used in this Training Module

Practice files you will use while working through the step-by-step exercises presented in this training guide can be [downloaded here](#). Finished solutions are available from the same download location as a reference for checking your work after completing the exercises on your own.

Setting up Internet Explorer to Use Siebel

1. Siebel must be added as a trusted site in Internet Explorer Options.
2. Trusted Sites security setting must be set to **Low**.
3. ActiveX controls must be allowed to run in Internet Explorer. This is turned on by default when the security setting is **Low**.

Note: On some systems the popup blocker must be set to allow pop-ups for the Siebel site. The settings required for Internet Explorer may vary depending on the individual Siebel installation.

Identifying Web Application Pages and Controls

Siebel Application Pages

The structure of the Siebel application uses multiple frames which house the active web pages. During navigation of the application, the frames are hidden and then come into focus as needed by the application. The same page may exist on multiple, different frames. When interrogating the application, using the **Create Global Web Page** option will ensure that the pages are matched without the frames. This way, no matter which frame currently shows the page, the page will be matched.

When you identify (via interrogation) a new web page, make sure to modify the **Document URL** match rule and use a **Query** string that is unique to identify that form as follows:

The screenshot displays the OpenSpan Studio interface with four main panels:

- Match Rules (Top Left):** Shows a list of targets. One target is highlighted with a red circle, containing the URL fragment: `ame=top_sweclient_sweclient_sweclient&SWEUID=1&SWETS= Si`.
- Match Rules (Top Right):** Shows a list of match rules. Two rules are checked: **Document Url Match Rule** and **Document Title Match Rule**.
- Selected Target (Bottom Left):** Shows the properties of the selected target. The **Misc** section includes:

ActiveLinkColor	0, 0, 255
BackColor	205, 213, 238
Cookie	SWEUAID=none; _sn=C
DefaultEncoding	windows-1252
Document ID	64BD008
Domain	siebel.openspan.com
Encoding	utf-8
Focused	False
ForeColor	0, 0, 0
IsAttached	True
- Selected Match Rule (Bottom Right):** Shows the configuration for the selected match rule. The **Query** section is highlighted with a red circle:

Control	Siebel_Call_Center1
Fragment	
Host	siebel.openspan.com
Path	/callcenter_enu/start.s
Port	80
Query	swescrbar
Culture	(User Culture)
IgnoreCase	True
Mode	Contains
Text	swescrbar

Siebel Application Controls - ElementID Match Rule

The Web adapter which is used for Siebel applications attempts to use the Element ID match rule, by default, in matching targets as this is the most efficient target matching method. However, it may be necessary to remove the Element ID match rule and use the Element Index and/or Element Inner Text match rules under the following circumstances:

- **Dynamic Element IDs** – for some Siebel controls, dynamic Element IDs are used by the application. This means that the Element ID changes each time the control is created. For example, dynamic Element IDs are used for controls in pop-up windows. Each time the window is re-opened, the IDs for the controls change.
- **Active Siebel Development Environments** – For environments where there are frequent updates to the Siebel version, do not use the Element IDs because these change with each Siebel rollout.

When Interrogating a Siebel target and the Element ID Match Rule is used, check to make sure the ID is static by navigating away from the control and then navigating back to it. If the control is still matched, the ID has not changed and the Element ID Match Rule can be used. However, if the Element ID Match Rule shows that the control is no longer matched, change the match rules for the target. Use the following steps to update match rules in these cases:

1. Set **UseElementId** to False in the properties for the control or select **Disable Use ElementID for this control** as show below:



2. Delete the Element ID match rule.
3. Add the Element Index match rule.
4. Set the Use Element ID property to False.
5. Add the Element Inner Text match rule.
6. Set the Use Element ID property to False.

This page intentionally left blank.

Chapter 2: Sample Project – Data Transfer

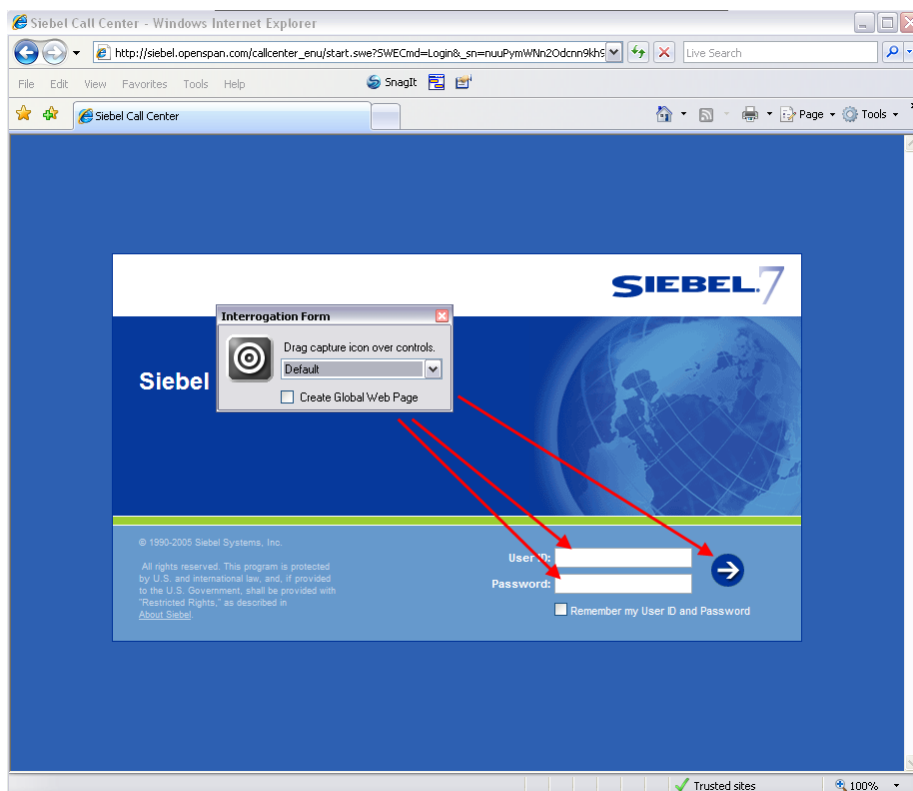
This project shows how to automate Siebel through OpenSpan Studio to:

- Login
- Get data from a list
- Initiate a process (make an appointment)

Automating the Login Process

The following automation sets up an automatic login to the Siebel training site used for this exercise.

1. Create a new solution called **Siebel Test**.
2. Add a Web Application Item (Web adapter). Name it Siebel.
3. Change the **StartPage** property to the login page for your Siebel system. For example, http://siebel.openspan.com/callcenter_enu/start.swe
4. Start Interrogation.
5. Interrogate the User Name and Password fields.
6. Interrogate the Login button.



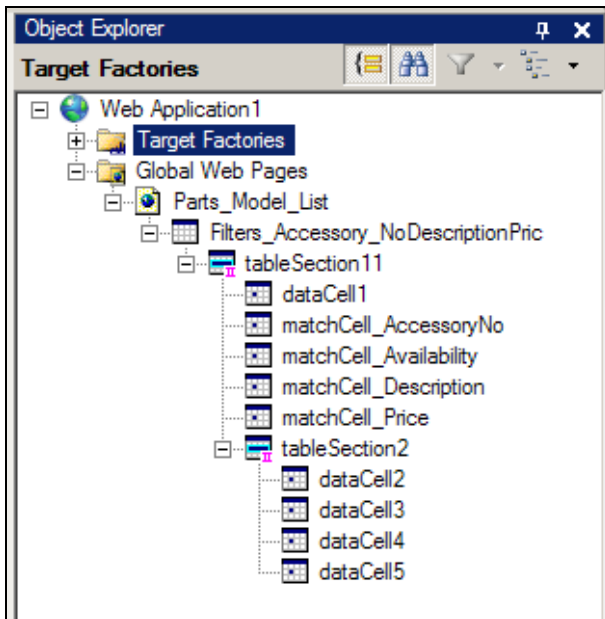
7. Stop Interrogation.
8. Save the solution.

Chapter 2: Sample Project – Data Transfer

9. Add a new automation. Name the automation **Login**
10. Complete the **Login** automation as shown below.



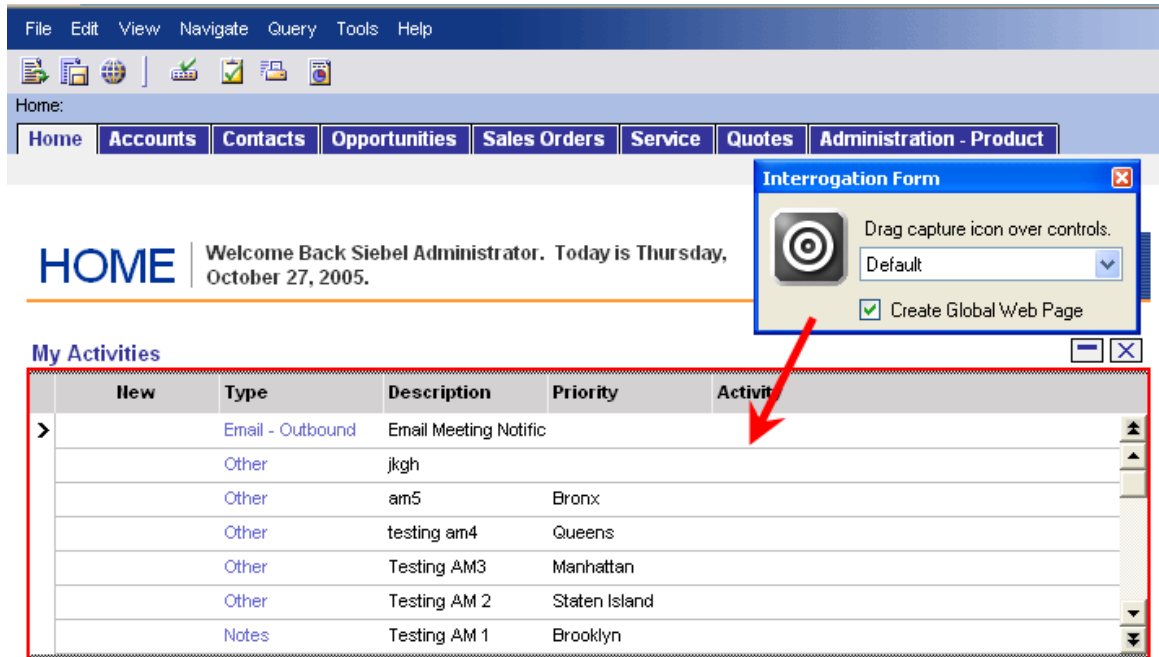
11. To automate the login, set the SWEUserName Text to the username you use to login (e.g., *sadmin*), and the SWEPassword Text to the password you use to login (e.g., *Siebel123*).
12. Save the solution.
13. Run the project. It should automatically login to the Siebel site.



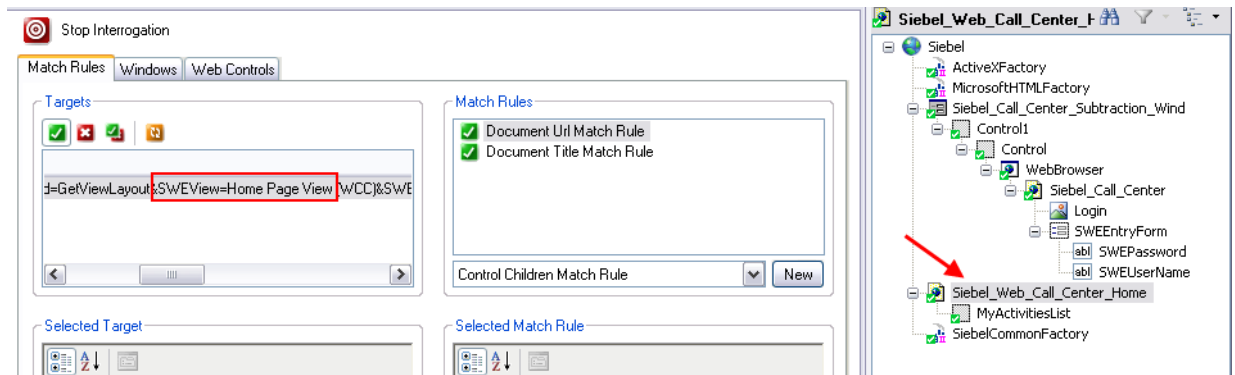
Get Data from Siebel List

Build the following automation that obtains data from the first page after initial login.

1. Start Interrogation.
2. Log in to Siebel. (User = **sadmin**; Password = **siebel123**)
3. With the “**Create Global Web Page**” Checked, Interrogate the “My Activities” List of the first screen and rename the control Siebellist1 to **MyActivitiesList**:

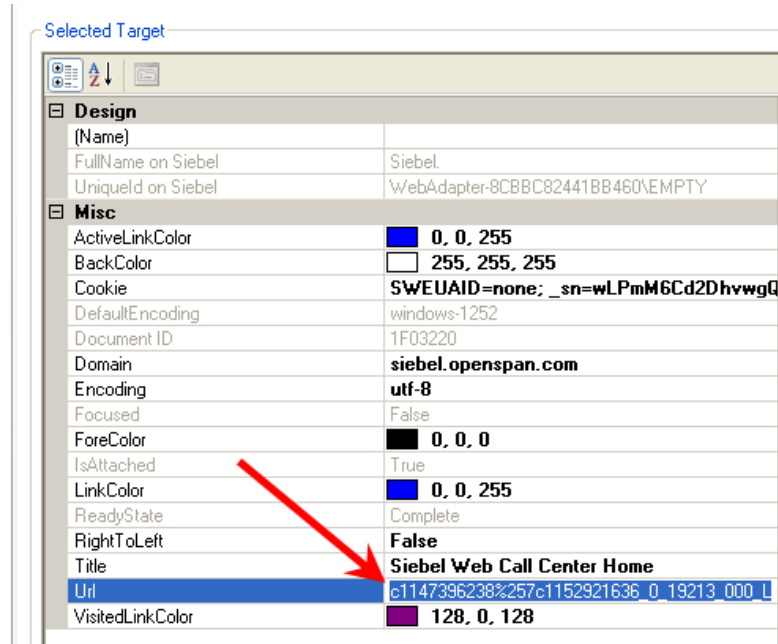


4. With the **Siebel_Web_Call_Center_Home** page selected within the Object Explorer, modify the **Document URL Match Rule** by adding a Query element to ensure this page is matched uniquely:

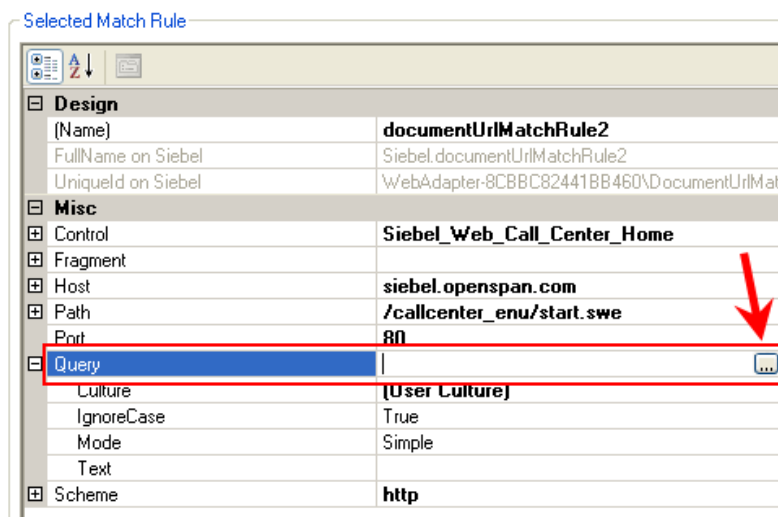


Chapter 2: Sample Project – Data Transfer

- Using the Targets pane, find a unique piece of text in the query section of the URL to uniquely identify the page. (E.g. SWEView=Home Page View)
- From the Selected Target pane, copy the complete URL string. This text is used to test our Regex match rule settings.



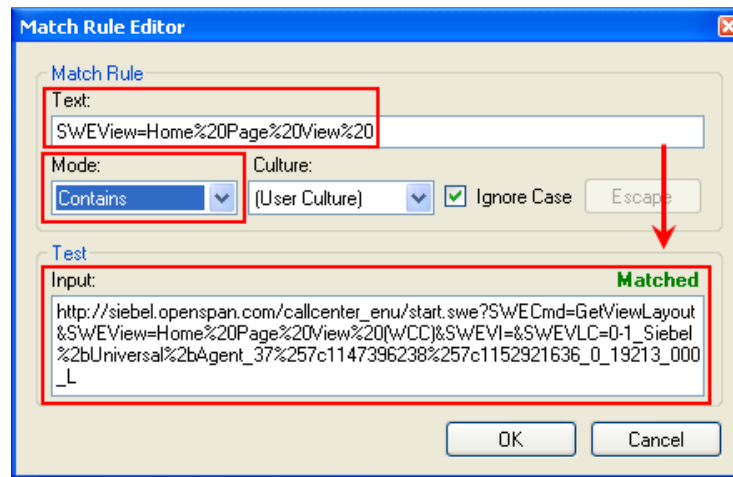
- Make sure you have the **Document Url Match Rule** selected in the **Match Rules** group box. From the **Selected Match Rule** groupbox expand the **Query** group by clicking on the +. Click into the Query area and click on the more button to open the **Match Rule Editor** dialog.



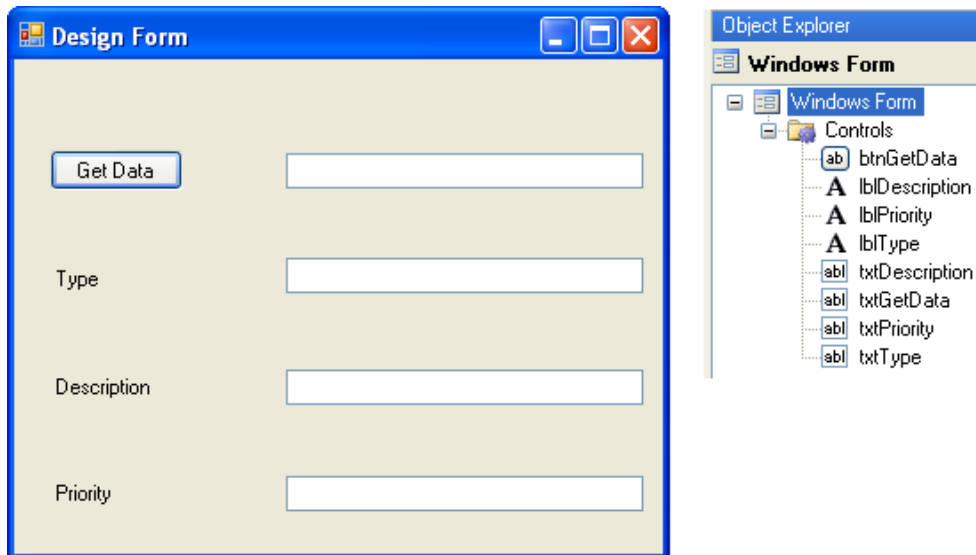
Chapter 2: Sample Project – Data Transfer

- d. Using the **Match Rule Editor** dialog, set and test the regex settings.
 - i. **Test Input field:** Paste the URL you copied in the step above.
 - ii. **Text Field:** `SWEView=Home%20Page%20View%20`
 - iii. **Mode:** Contains

NOTE: Notice that the Test is matched correctly. Now click the OK button the Match Rule Editor to accept your changes.



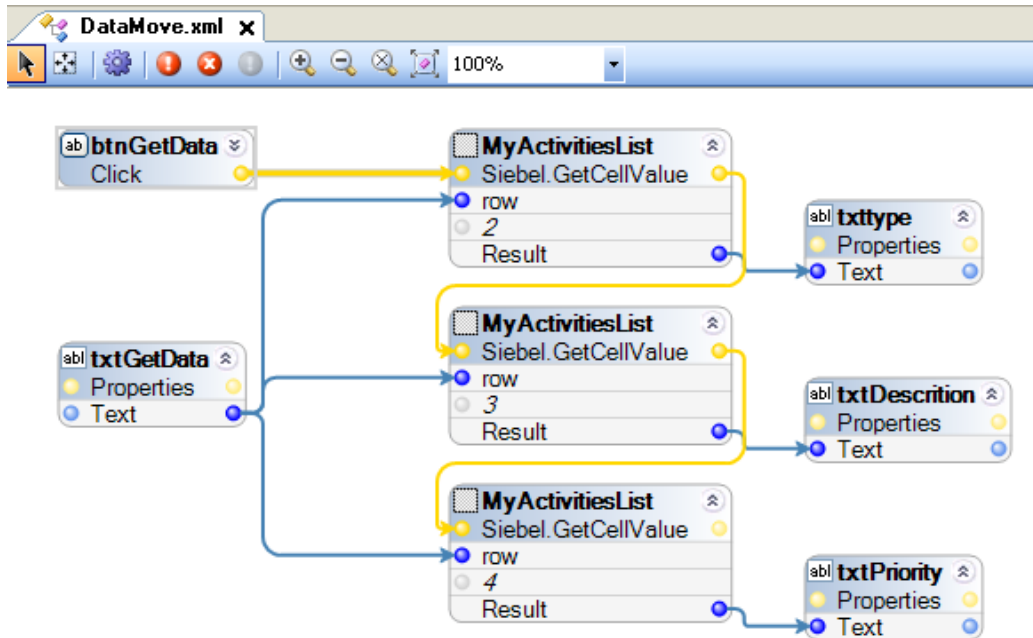
5. Stop Interrogation and create a windows form as follows:



6. Label the items as shown in the Object Explorer image above.
7. Create a new automation. Name it **DataMove**.

Chapter 2: Sample Project – Data Transfer

- This automation retrieves the information in the second, third and a fourth column for a given row in the Siebel list and outputs the values to textboxes on the Windows form. To retrieve data from these columns, the Column parameter on the **SiebelList Siebel.GetCellValue** method is set to 2, 3, and 4 respectively as shown in the following automation image. Complete the DataMove automation as follows:



- Save and run the project.
- Once the login automation is complete and the Home page displays, enter a row number into the Get Data text field on the Design Form. Press the Get Data button. An example of the active Design Form is shown below:

The Design Form window displays the results of the automation. It contains a 'Get Data' button and four textboxes with the following values:

Get Data	3
Type	Other
Description	am5
Priority	Bronx

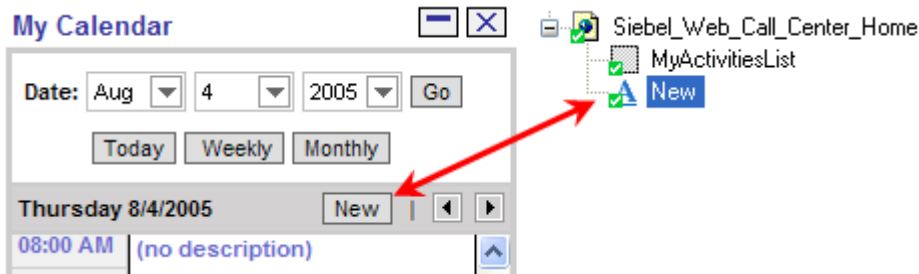
- Stop the project.

Initiate a Process

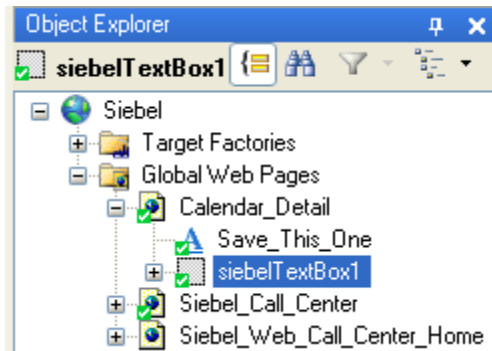
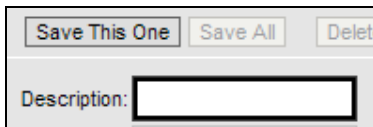
The following steps create an automation which adds an appointment to the calendar shown on the first page of the Siebel system.

Perform the following steps:

1. Create a new automation. Name it **MakeAppointment**.
2. With the “**Create Global Web Page**” **Checked**, start the Interrogator.
3. Log in to Siebel.
4. Interrogate the **New** button in the Calendar area. You should now have a “**New**” link element in the Object Explorer under the same page as the “**MyActivitiesList**”.

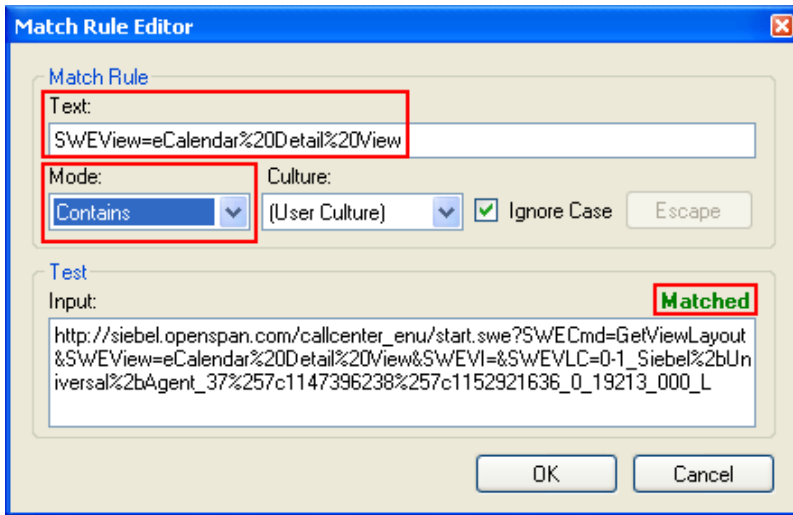


5. Click the **New** button to move to the Calendar Detail Tab.
6. Interrogate the **Description Field** and the **Save This One** Button.



Chapter 2: Sample Project – Data Transfer

7. Modify the Document URL Match Rule for the **Calendar_Detail** page to ensure that the page is uniquely matched. Modify the query string of the Document Url Match Rule. (e.g. [SWEView=eCalendar%20Detail%20View](#)):



The Match Rule Editor dialog box is shown. The 'Text' field contains the query string 'SWEView=eCalendar%20Detail%20View'. The 'Mode' dropdown is set to 'Contains'. The 'Culture' dropdown is set to '(User Culture)'. The 'Ignore Case' checkbox is checked. The 'Test' section shows the 'Input' field with a long URL, and the result 'Matched' is displayed in a red box. The 'OK' and 'Cancel' buttons are at the bottom.

8. Stop Interrogation.
9. Modify the Design Form to include a “Make Appointment” button and another text box.

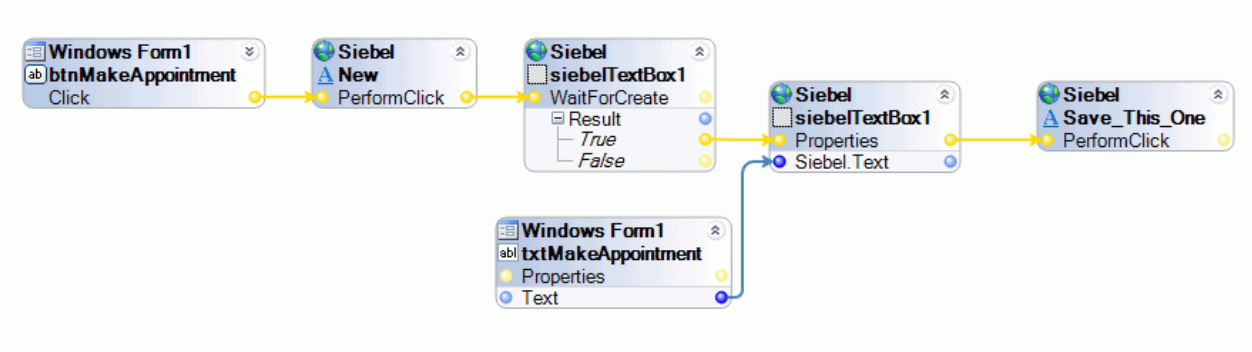


The Design Form dialog box is shown. It contains a 'Get Data' button and a text box. Below these are labels for 'Type', 'Description', and 'Priority', each followed by a text box. At the bottom, there is a 'Make Appointment' button and another text box.

10. Name the Make Appointment button **btnMakeAppointment**.
11. Name the new text box **txtMakeAppointment**.

Chapter 2: Sample Project – Data Transfer

12. Complete the MakeAppointment automation as shown below:

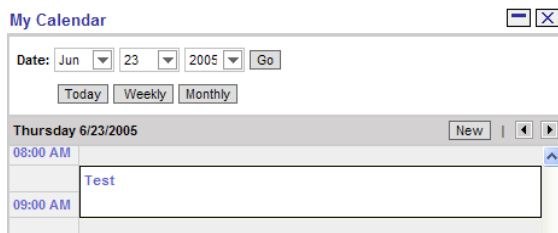


13. Save and Run the solution.

14. Wait for the Login automation to run.

15. Enter a description into the text field next to the Make Appointment button on the design form.

16. Press the Make Appointment button. The application will navigate to the Calendar tab, enter the appointment, and then return to the Home tab. See the following result; however remember the entry will be for the current time of day:

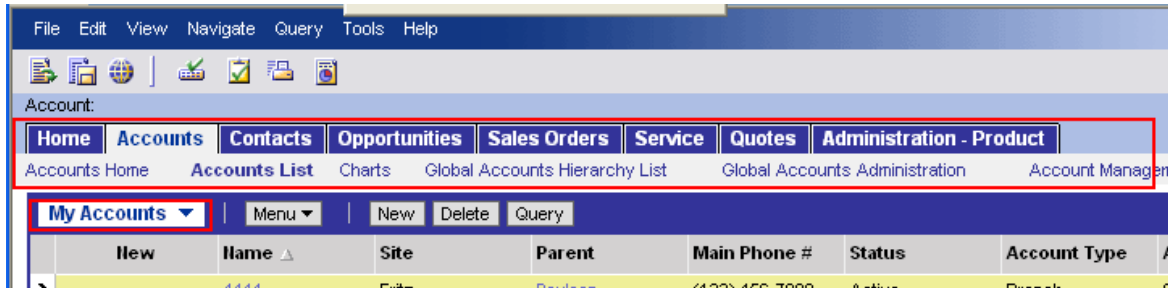


17. Stop the project.

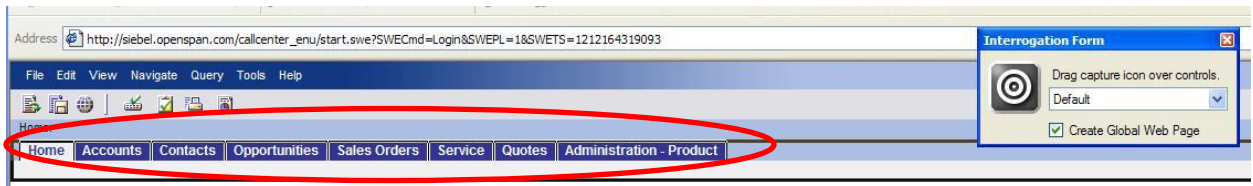
This page intentionally left blank.

Chapter 3: Navigation Controls

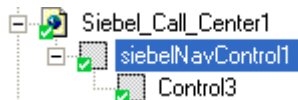
There are two different visual controls that OpenSpan identifies as a SiebelNavControl. The first being the Navigation tabs and the other being a combobox with the white backing.



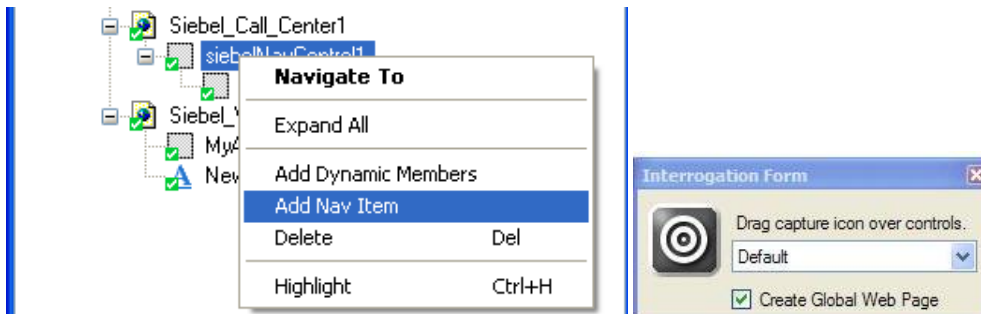
When adding navigational tabs, you will not be able to interrogate the individual tabs.



You must interrogate the entire Navigation control (as shown above). Once the control is added it appears as a SiebelNavControl in the object explorer:

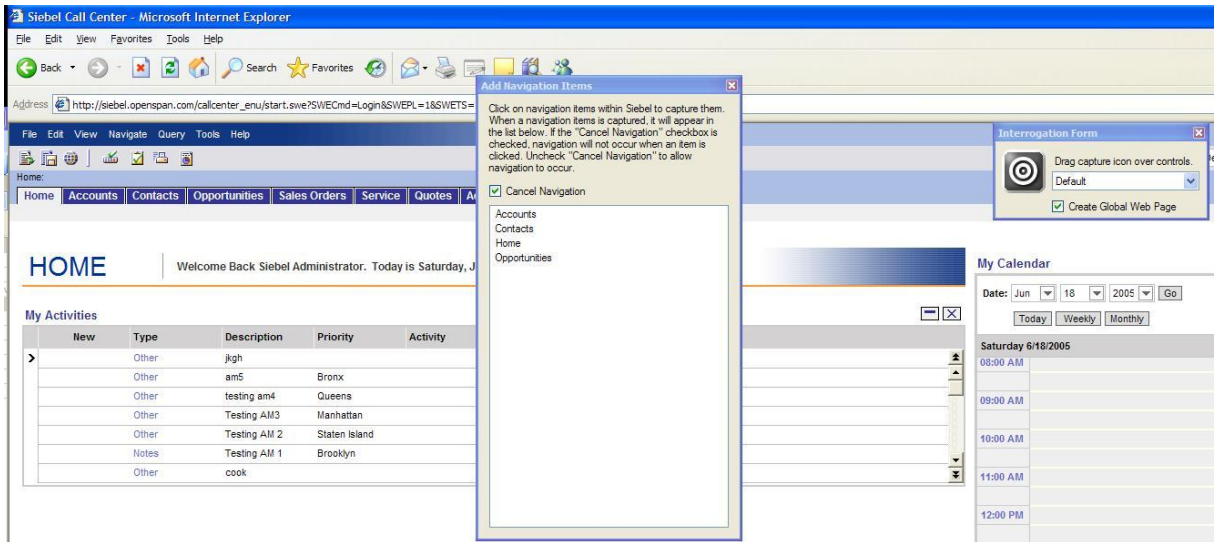


Right-click on the SiebelNavControl and select the Add Nav Item option from the context menu.

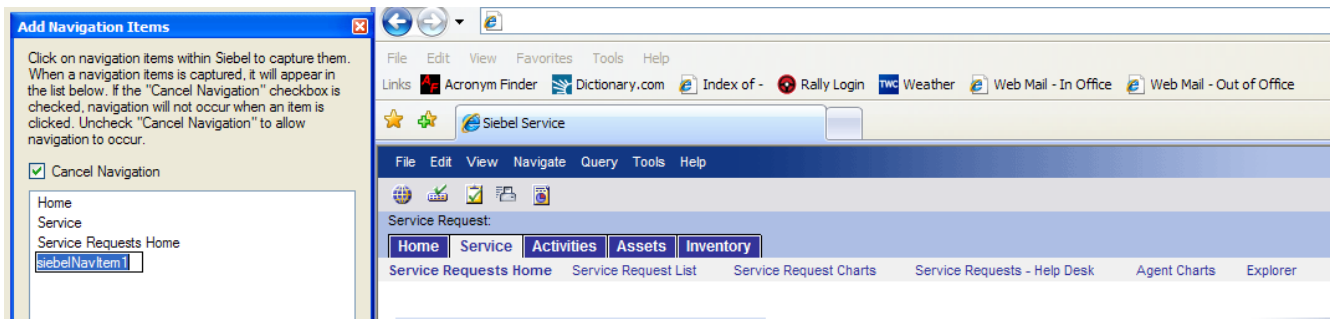


The Add Navigation Items dialog displays. Click on the desired tab button in Siebel and then name it appropriately in the Add Navigation Items window to create the navigation objects. During the creation of the navigation items, checking the "Cancel Navigation" checkbox stops the Siebel application from taking any action while allowing OpenSpan to capture the link. You must uncheck the "Cancel Navigation" checkbox if you want Siebel to navigate to the desired link.

Chapter 3: Navigation Controls



To interrogate a sub menu item, uncheck the **Cancel Navigation** checkbox to allow Siebel to navigate to a menu items page. Recheck the **Cancel Navigation** checkbox and click each desired sub menu item to add them to the list.

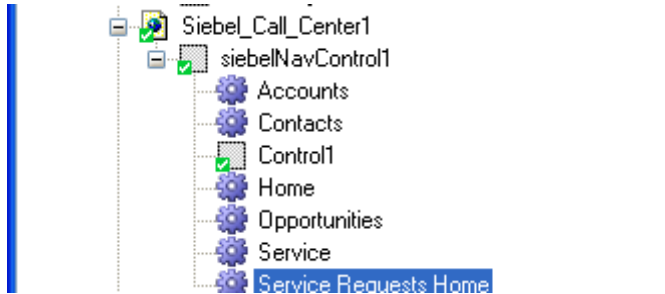


When you close the Add Navigation Items dialog, each selected navigation object then appears under the SiebelNavControl in the Object Explorer.



Chapter 3: Navigation Controls

Sub menu items are added to the list just as menu items. They do not appear in a subordinate format in the Object Explorer. Sub Menu items may be directly called within an automation.

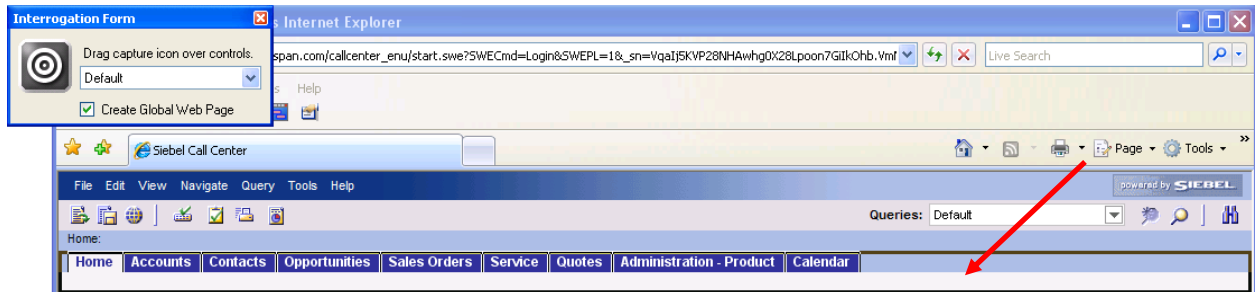


Exercise – Automating Navigation

Use the steps which follow to create a project which automatically navigates to the Contact List page.

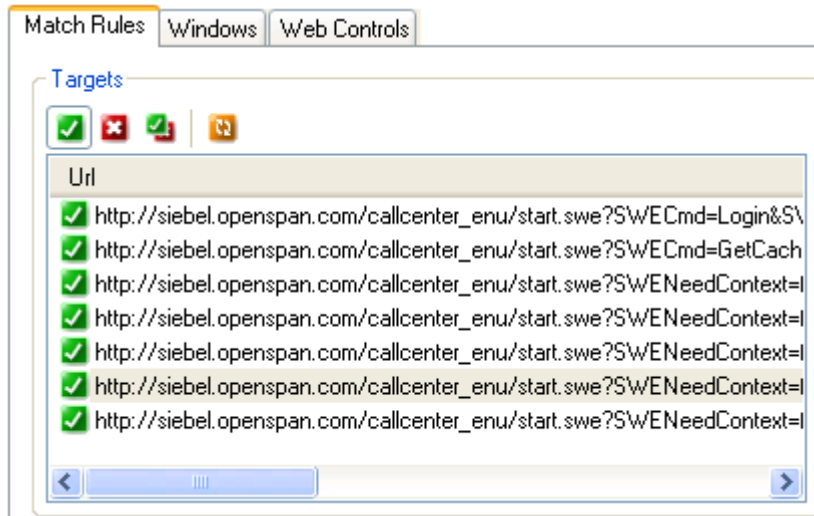
Create Navigation Controls

1. Return to the Siebel Test solution you completed in the earlier exercise.
2. Start the Interrogation of the Siebel application. Login to the application to open the Home page.
3. With the “Create Global Web Page” checked; Interrogate the Navigation bar near the top of the window:



Chapter 3: Navigation Controls

4. Check the matching targets for the page (Siebel_Call_Center1) associated with the Navigation bar:



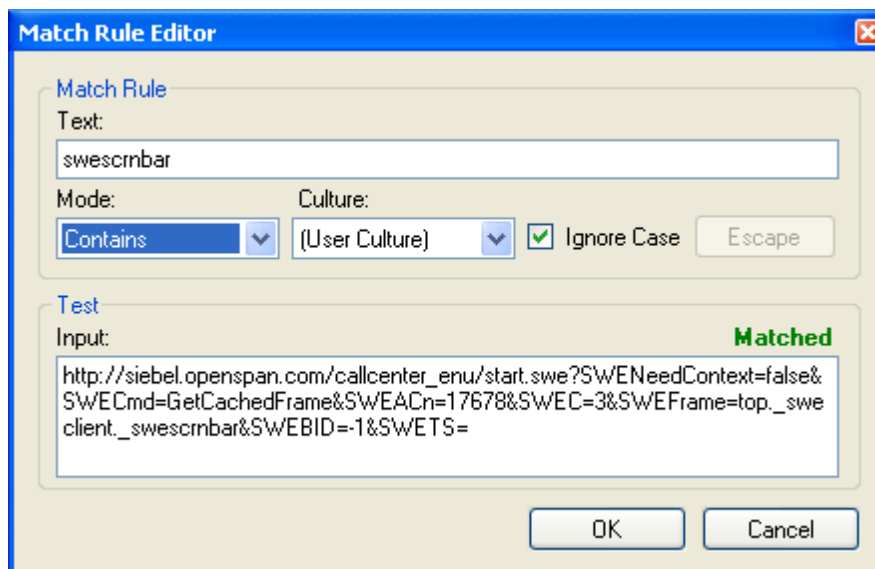
Note that multiple pages match the target. **Make sure that you do NOT click on another URL within the Targets window when attempting the next step.**

5. Modify the Query portion of the Document URL Match Rule to uniquely match this page. The full URL for the page is:

```
http://siebel.openspan.com/callcenter_enu/start.swe?SWENeedContext=false&SWECmd=GetC  
achedFrame&SWEACn=12033&SWEC=2&SWEFrame=top._sweclient._swescrnbar&SWEBID=-  
1&SWETS=
```

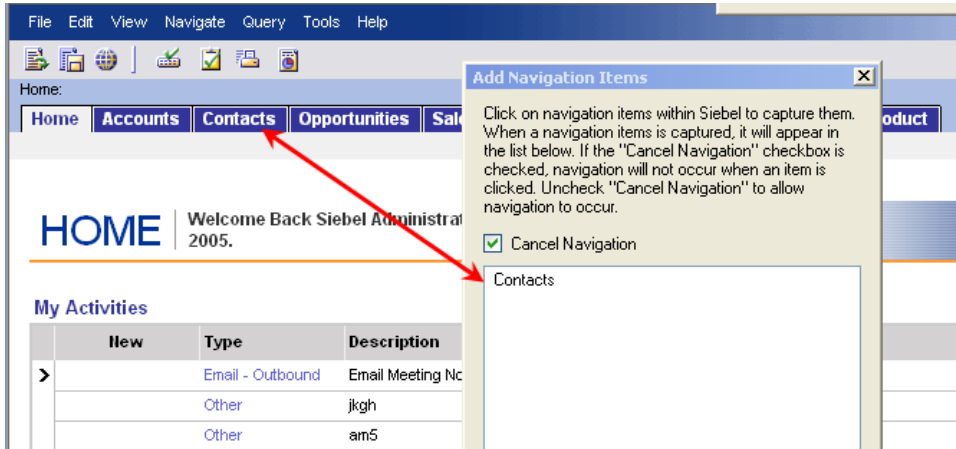
The unique portion is “swescrnbar”. So we can modify the **Mode** property for the Rule to indicate that the Text must contain the text string **swescrnbar**. Click on the **Query** property for the Document URL Match Rule to open the editor.

6. Complete the **Match Rule Editor** as shown below:

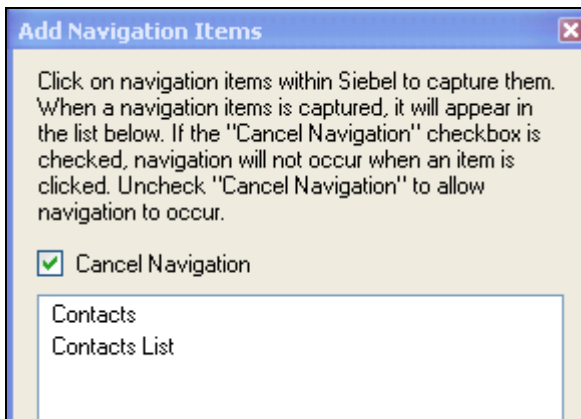


Chapter 3: Navigation Controls

7. Click **OK** to close the Editor and then click the **Refresh** matching button. Only the desired page should appear in the **Targets** pane.
8. Highlight the SiebelNavControl object in the **Object Explorer**, right-click and select **Add Nav Item**. The **Add Navigation Items** dialog displays.
9. Select the **Cancel Navigation** checkbox. Click on the Contacts link in Siebel and then rename the object created in the **Add Navigation Items** dialog as **Contacts**.

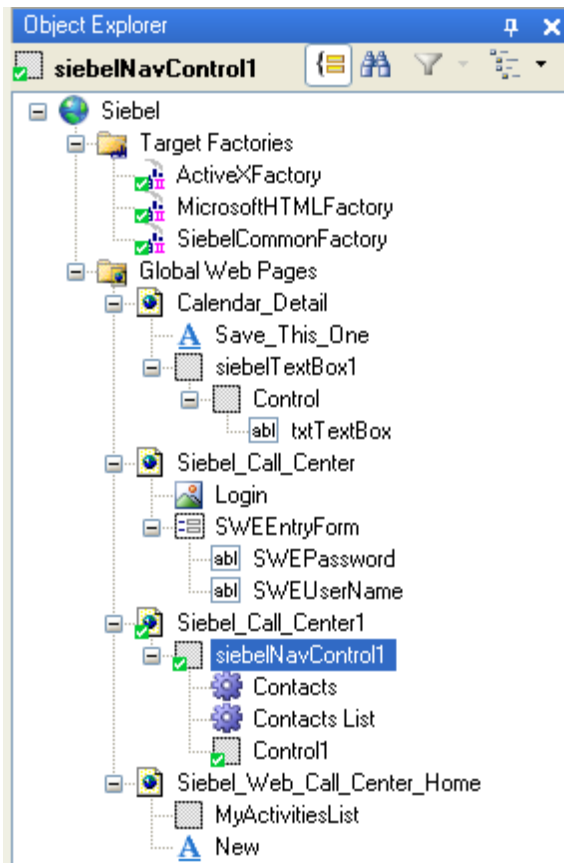


10. Clear the **Cancel Navigation** checkbox. Navigate to the Contacts page by selecting the Contacts link in the application.
11. Check the **Cancel Navigation** checkbox. Click on the Contacts List link and rename the added link to **Contacts List**. Your **Add Navigation Items** dialog should look like the following:



Chapter 3: Navigation Controls

12. Close the **Add Navigation Items** dialog. The links appear in the **Object Explorer** under the SiebelNavControl:

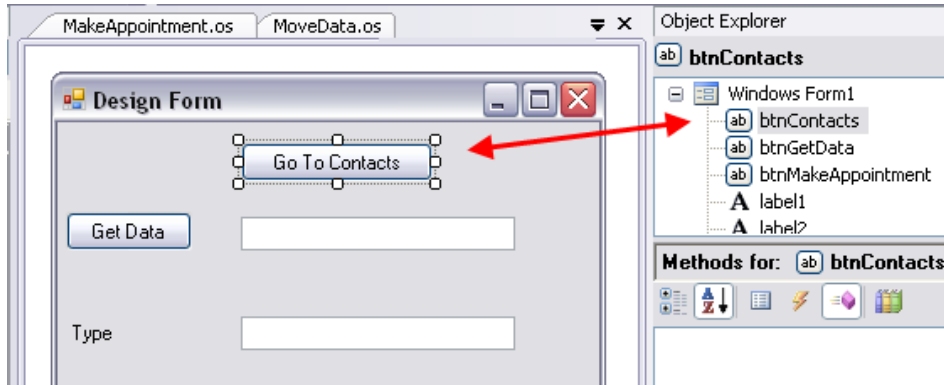


13. Stop the Interrogator.

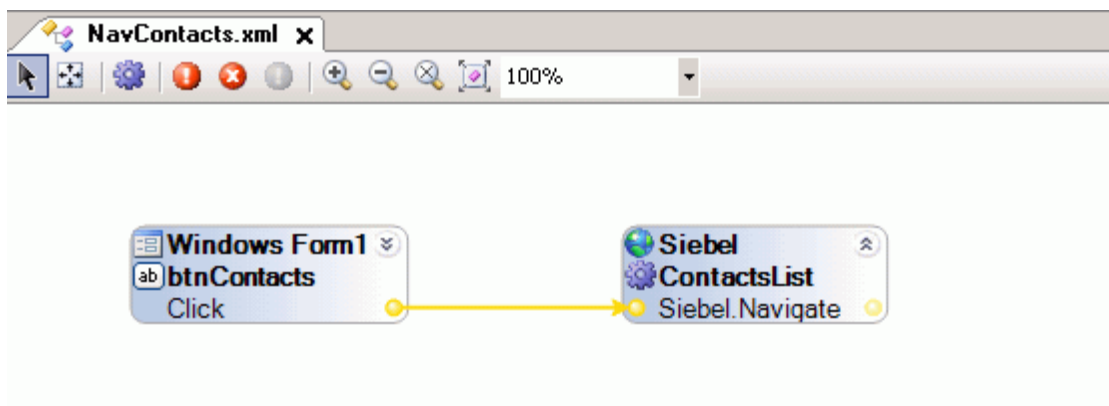
Complete Automate Navigation Solution

The following steps create an automation that will automate navigating to the Contacts Lists page.

1. Modify the Design Form. Add a button near the top of the form. Name the button **btnContacts** and set the Text property to **Go To Contacts**.



2. Create a new automation: **NavContacts**.
3. Connect the **btnContacts** Click event to the **Contacts Lists** Navigate method. The Complete automation is shown below:



Note: You do not have to first navigate to the Contacts page and then invoke the Contacts List link.

4. Save and run the project.
5. Once the Home page has loaded, click the Go to Contacts button on the Design form. The Contact List will load.

This page intentionally left blank.

Chapter 4: Working with Pop-up Dialog Boxes

Siebel contains popup dialogs accessible from multivalue and SiebelList controls. The following exercise demonstrates working with a popup dialog.

Exercise – Popup Dialog

This exercise pushes data from the Windows form used in the previous exercise to the Contacts page in Siebel to create a new contact. Part of the process is assigning an account representative for the contact. This is done by selecting the account representative in a popup dialog accessed from a multivalue control.

1. Open the solution used in the previous exercise and save it as **Siebel Test 3**.
2. Open the Windows form and modify it as shown below:

The screenshot shows a 'Design Form' window with the following controls and callouts:

- Get Data**: A button on the top left.
- Go To Contacts**: A button on the top right.
- Type**: A text input field.
- Description**: A text input field.
- Priority**: A text input field.
- Last Name**: A text input field with callout **txtLastName**.
- First Name**: A text input field with callout **txtFirstName**.
- Accounts**: A list box with callout **lbxAccounts**.
- Make Appointment**: A button on the bottom left.
- Create Contacts**: A button on the bottom right with callout **btnCreateContacts**.

Note: The names of the controls are as shown in the image above. Use the names of your account representatives in the List box.

3. Save the solution.
4. Open the web adapter for the Siebel application and start the Interrogator.
5. Login and navigate to the Contact List page.

Chapter 4: Working with Pop-up Dialog Boxes

6. With the **Create Global Web Page** checked, Interrogate the New button/link near the bottom of the page. Also interrogate the **Last Name**, **First Name**, and **Account** controls. Name the controls as you add them:
 - a. Last Name = SiebelTxtLastName
 - b. First Name = SiebelTxtFirstName
 - c. Account = SiebelMultiValueControl1Account

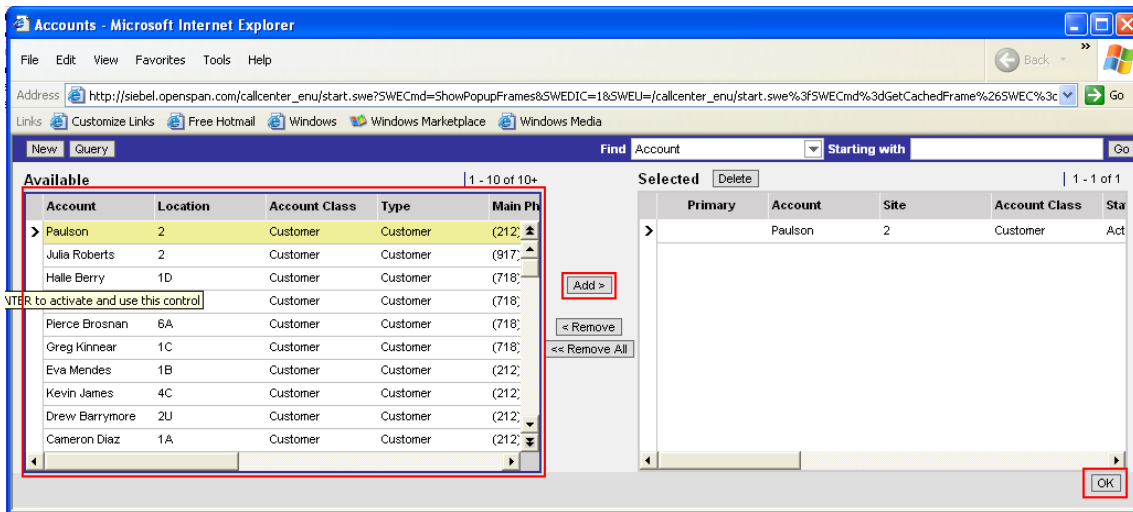
Note: All Siebel specific controls are identified in the **Object Explorer** with a prefix **Siebel**. When interrogating a textbox for example, the **Object Explorer** shows three controls are added: **SiebelTextBox1→Control1→txtTextBox1**. The only control what should be used within an automation is the Siebel Component.

The screenshot displays the Siebel Call Center interface within a Windows Internet Explorer browser. The browser address bar shows the URL: `http://siebel.openspan.com/calcenter_enu/start.swe?SWECmd=Login&SWEPL=18&SWETS=1221235648125`. The interface includes a navigation menu with tabs for Home, Accounts, Contacts, Opportunities, Sales Orders, Service, Quotes, and Administration - Product. Below the menu is a 'My Contacts' section with a table of contact records. The table has columns for Last Name, First Name, Mr/Ms, Account, Work Phone #, Personal Address, City, State, Postal Code, and Email. The first row is highlighted in yellow and contains the following data: Last Name: Administrator, First Name: Siebel, Mr/Ms: (blank), Account: 13133, Work Phone #: (blank), Personal Address: New City, State: (blank), Postal Code: Fun in the sun, Email: sadmin@siebel.com.

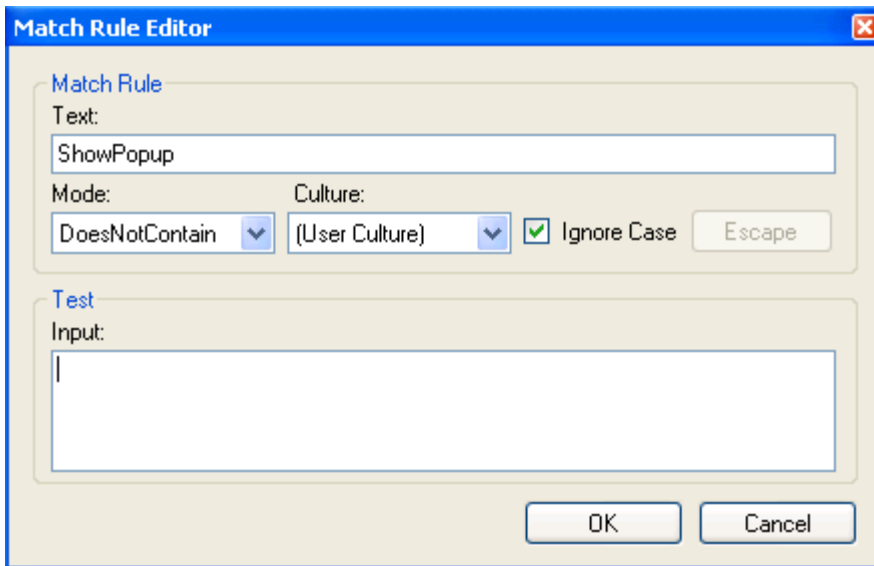
Below the table is a 'Siebel Administrator' form with a 'New' button highlighted in red. The form contains several input fields, with 'Last Name' and 'First Name' highlighted in red. The 'Last Name' field contains the text 'Administrator' and the 'First Name' field contains the text 'Siebel'. Other fields include 'Work #', 'Main Fax #', 'Mobile Phone #', 'Account', 'Site', 'Address' (13133), 'City' (New City), 'State', 'Job Title' (Sys Admin), 'Mr/Ms' (dropdown), 'Email' (sadmin@siebel.com), 'Zip Code' (Fun in the sun), and 'Country' (dropdown).

Chapter 4: Working with Pop-up Dialog Boxes

- Click on the popup button in the Account control to open the Accounts page. Interrogate the Available list, Add and OK button/link. Rename the SiebelList to SiebelListAccount the other control names are fine.



- In the Object Explorer, highlight the Accounts page. Note that multiple pages are matched in the Targets pane. To correct this situation, modify the match Document URL Match Rule. The URL for the page that needs to be matched is the one that **does not contain** the string "ShowPopup" in the Query section. Your Match Rule Editor should look like the following:

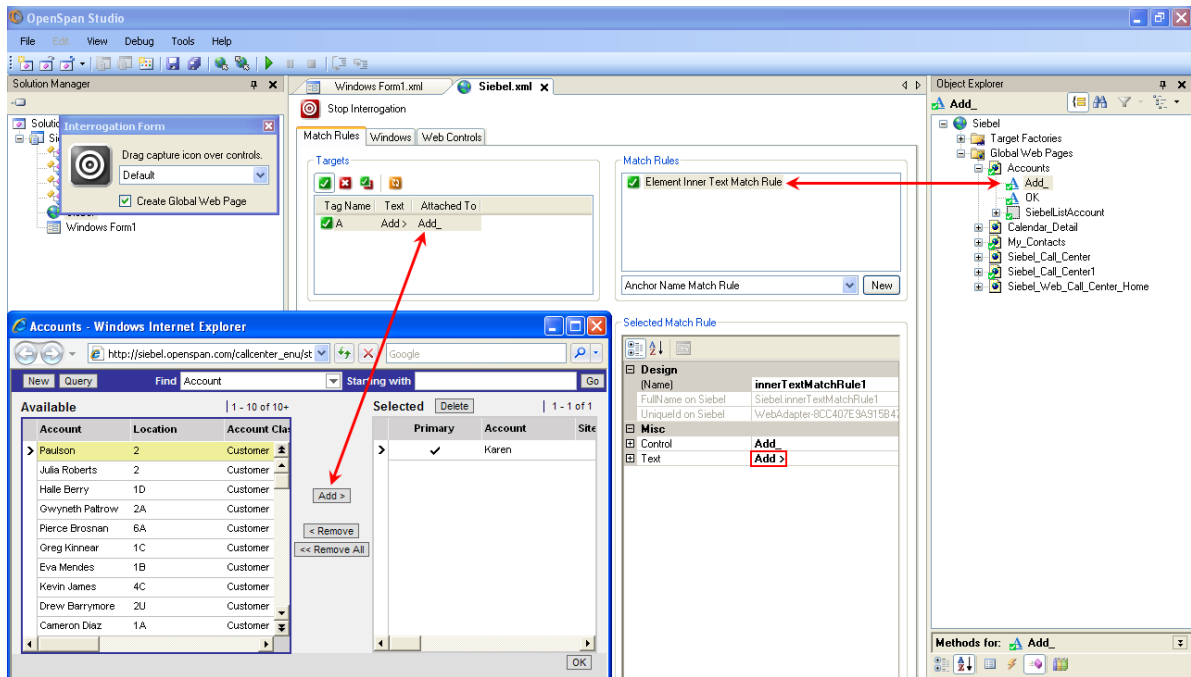


- Click OK to save the Match Rule Editor entries and then click the Refresh Matching button. Only one page should match:

http://siebel.openspan.com/callcenter_enu/start.swe?SWECmd=GetCachedFrame&SWEC=10&SWEFrametop._sweclient._swecontent._sweview._swetmp&SWESP=1

Chapter 4: Working with Pop-up Dialog Boxes

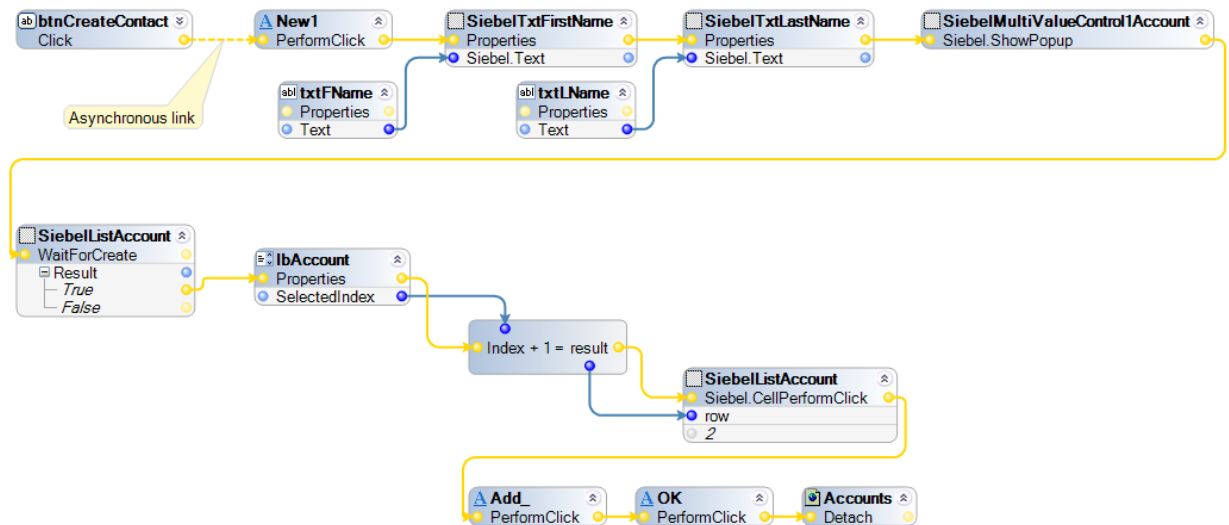
- For the Add button/link, change the match rule from Element ID to Element Inner Text Match Rule. (e.g. Text = Add >). Make sure you set the "UseElementID" property to False after you delete the ElementID Match Rule.



- For the OK button/link, change the match rule from Element ID to Element Inner Text Match Rule. (e.g. Text = OK). Make sure you set the "UseElementID" property to False after you delete the ElementID Match Rule.

- Stop the Interrogator.

- Create a new automation: CreateContact. Complete the automation as shown below:



Chapter 4: Working with Pop-up Dialog Boxes

Automation Notes:

- The `siebelListAccount.WaitForCreate` method is required as the automation must ensure the `SiebelListAccount` is created before calling the `Siebel.CellPerformClick` method.
 - The Numeric Expression is required in this automation as the Index of the Listbox starts at the value of 0 and the Row values for the `SiebelList2` start at 1.
 - The behavior of the Siebel Popup windows is to be moved off screen when closed not destroyed. It is only destroyed by the action of bring the window back into focus. (For example, calling the `ShowPopUp` method or clicking in Siebel.) To work around this behavior we must use the “Detach” method which tells the OpenSpan Platform that the screen has been destroyed. Now when Siebel creates a new one and brings it into focus we can see the creation.
14. Save and run the project. Once the login page has passed, click the Go To Contacts button on the Windows Form. When the contacts list displays, enter a name in the Last/First name fields on the Windows Form and select an Account Manager name. Next, click the Create Contact button. The Last Name and First Name fields should populate in Siebel and then the Accounts page should open. Your manager should automatically be selected and the Accounts page closes. The manager name will appear in the Account field.