OpenSpan Core Training

OpenSpan Studio - Web Integration:

- CHAPTER 1: Defining the Web Adapter Design Properties
- CHAPTER 2: Interrogating Web Applications
- CHAPTER 3: Understanding Web Adapter Match Rules
- CHAPTER 4: Web Application Automations
- CHAPTER 5: Web Application Navigation

Introduction 5
Prerequisites 5
Conventions 5
CHAPTER 1: Defining Web Adapter Properties, Methods and Events 7
Windows and Web Adapter Properties, Methods and Events 7
StartPage Property 8
Target Creation 8
Created Event 8
IsCreated Property 9
WaitForCreate Method 9
CHAPTER 2: Interrogating Web Applications 11
Exploring the Interrogation Function 11
Select Element 11
Create Global Web Page 13
Exercise – Global Web Page 15
Interrogated Objects 17
Parts of the Object Explorer Hierarchy 17
Create Control for Hidden Target 18
Exercise - Interrogating Web Applications 19
Select Element 20
Create Hidden Control 21
CHAPTER 3: Understanding Web Adapter Match Rules 25
Web Adapter Match Rules 25
Web Application Match Rules 26
ActiveX Factories – Module Name Match Rule 26
IE Frame - Window Text and Class Name Match Rules 26
Window Text Match Rule 27
Window Class Name Match Rule 27
TabWindowClass - Window Text and Class Name Match Rules 27
Shell DocObject View - Window Style Match Rule 28
Browser – Window Class and Window Style Match Rules 30
HTML Page - Web Object Matching 30
Default Match Rules 31
Exercise - Attribute Value Match Rule 34
Key Default Match Rules 37
Modifying Match Rules 39
Document Title and Document URL Match Rules 39
Exercise - Modifying Document Title Match Rule 40
Ambiguous Matching 47
Controls 47
Exercise - Modifying Element Inner Text Match Rule 48
CHAPTER 4: Web Application Automations 53
Automating Login 53
Exercise – Login Form 53
Create Login Window 54
Add Login Automations 55

Test Login Automation - Part 1 64
Add Web Application Verification Logic 64
Test Login Automation - Part 2 69
Add Login Automation Logic 69
Exercise – Raising Web Control Events 72
Test Login Automation Part 3 76
CHAPTER 5: Web Application Navigation 77
Interrogating Links 77
Exercise – Interrogate Links 77
Create Global Web Page for Links 77
Automate Link Selection 79
Navigating Links 81
Final Exercise - Completing the Auto-Login Process 83
Completed Solution 83

INTRODUCTION

This course provides an in-depth look at how to use the OpenSpan Studio Web Adapter to integrate web applications with your solutions. In the Basics training, you learned how to interrogate a web application and how to create automations that execute data and event logic with web objects. In this course, you will learn the fine points of interrogating web applications, modifying match rules, and navigating web pages.

Prerequisites

This course requires a working knowledge of OpenSpan Studio, including familiarity with general interrogation concepts and the ability to create a basic web-based solution using OpenSpan Studio.

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

- OpenSpan Studio 4.5
- Internet Explorer version 6, 7, or 8.
- Successful completion of the **OpenSpan Studio Basics** training module
- Training project files available for download from the OpenSpan Community website Learning and Certification page
- Access to the OpenSpan training website: (http://training.openspan.com/index.html)

Conventions

This document uses the following typographical rules and conventions:

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.**
Remember	**Definitions of terms and important concepts that bear remembering.**
TIP	**Next to the Tip icon, you can find best practices and shortcuts to use OpenSpan Studio more effectively.**

This page intentionally left blank.

CHAPTER 1: Defining Web Adapter Properties, Methods and Events

The OpenSpan Studio Web application adapter enables you to integrate web applications in OpenSpan projects. Note that OpenSpan supports web applications running in Internet Explorer, versions 6, 7 and 8. You can create OpenSpan projects that integrate web applications with other applications, automate tasks in a web application, and monitor events for a web application. This chapter provides an overview of the following:

- **StartPage** property
- **Created** event
- **IsCreated** property
- **WaitForCreate** method

Windows and Web Adapter Properties, Methods and Events

Design Properties

The table below compares the base design properties for the two adapters:

Windows Adapter Properties	Web Adapter Properties
(Name)	(Name)
▨	AddressBarVisible
Arguments	▨
CloseTimeout	CloseTimeout
Extender	Extender
Full Name	Full Name
HideApplicationAtRuntime	HideApplicationAtRuntime
HookChildProcesses	HookChildProcesses
IdleTimeout	IdleTimeout
▨	IgnoreMainBrowser
IsRunning	IsRunning
IsStartStoppable	IsStartStoppable
IsStopping	IsStopping
▨	MenuBarVisible
Path	▨
Folder	▨
ResolvePath	▨
SendMessageTimeout	SendMessageTimeout
StartMethod	▨
StartOnProjectStart	StartOnProjectStart
▨	StartPage
StartTimeout	StartTimeout
Stop Method	StopMethod

Windows Adapter Properties	Web Adapter Properties
▨	StopWhenMainWindowsCloses
▨	SubscribeToWebBrowserEvents
SuppressForegroundWindows	SuppressForegroundWindows
TargetPath	▨
TerminateTimeout	TerminateTimeout
▨	ToolBarVisible
UniqueID	UniqueID
WorkingDirectory	▨

As you can see from the previous table, many of the properties are shared between the adapters. The main difference being the way you identify the target application for the adapter. With the Web Adapter, you use the **StartPage** property and with the Windows adapter you use the **Path** or a combination of the **Path** and **TargetPath** properties.

StartPage Property

The **StartPage** property specifies the first .html page to open for the application. The property accepts the site location in multiple formats. For example:

- www.openspan.com
- http://216.215.233.67/
- Z:\Inetpub\wwwroot\index.htm

If you are creating a solution that will navigate a browser, you can add a web application to your solution and leave the **StartPage** property blank. In this case, the adapter will open the browser and navigate to your home page.

The **StartPage** property sets the page that will load when the adapter is started.

Target Creation

As you navigate a web application, you will notice that the page loads in the browser and that the web controls are painted on the page. When automating actions that interact with web pages and controls, it is important to ensure that the target pages and controls are matched ***before*** your automation sends data or events to the controls. OpenSpan Studio provides properties, methods, and events to ensure targets are matched.

Created Event

When OpenSpan Studio matches a target, the **Created** event for the object is triggered. For example, in the simple *Demo* solution you created in the OpenSpan Studio Basics training, when the Google page is fully loaded in the browser, a **Created** event is triggered based on the search text box (**q**) matching. Similarly, a **Created** event is triggered when the **Search** button is matched. If you navigate from the Google page, the interrogated objects (that is, the textbox and button) are no longer matched.

IsCreated Property

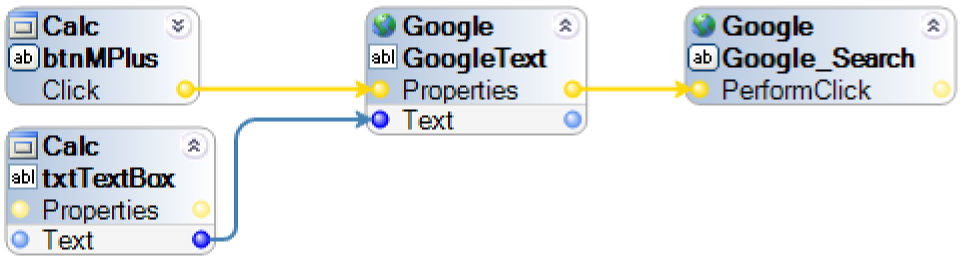
This property is **False** until the **Created** event has been triggered for the object. Returning to the example of the *Demo* solution, the search text box **IsCreated** property would be false until the Google page is loaded and matched by the OpenSpan Studio match rules for the object. When you navigate away from the Google page, the objects are no longer matched and the **IsCreated** property is false.

WaitForCreate Method

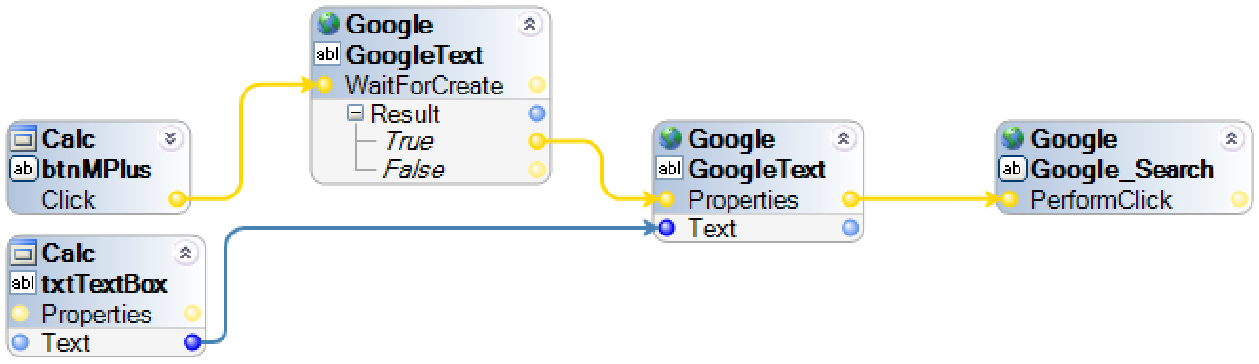
This method enables you to setup automation logic to wait until the **Created** event has been triggered for an object before proceeding to the next step of the automation. All objects have an *implicit wait for create*.

This means that OpenSpan Studio will wait until the **Created** event is raised for an object before attempting to use the object in automation logic. The implicit wait interval is set to 30000 milliseconds by default. After 30000 milliseconds, the next step in the automation is attempted.

The following automation demonstrates the *Demo* solution you created in the *OpenSpan Studio Basics* training module:



When the user clicks the **btnMPlus** (**M+**) button on the Calculator, OpenSpan Studio triggers the **GoogleText** textbox on the Google page to pull the text from the from the Calculator **txtTextBox** (Result) textbox. In order for this automation logic to occur, the **GoogleText** textbox must be matched (created). OpenSpan Studio will wait 30 seconds for the **GoogleText** textbox to respond to the click input event before continuing to the next step, which is the **PerformClick** method for the **Google_Search** button. The following example performs the same logic:



The above automation uses the **WaitForCreate** method within the logic flow. By adding this property, the designer can provide a path for the automation logic to follow in the case that the **GoogleText** textbox is not matched within 30 seconds of the **btnMPlus (M+)** button click. By using the **WaitForCreate** method, the solution designer can specify the actions taken by the solution should an object fail to match at runtime.

In web applications, the loading of pages and web controls dictates when objects are matched. In chapter 3, you will get a chance to view the matching process in detail. Since the speed at which web application pages and objects are loaded in a browser will vary, you need to consider target creation in your web application solutions. As you work through the remaining chapters in this course, you will get practice using the **WaitForCreate** method, **Created** event, and **IsCreated** property in your solutions.

CHAPTER 2: INTERROGATING WEB APPLICATIONS

After setting the **StartPage** for your web application, you are ready to interrogate the application. This chapter covers the following topics related to web application interrogation:

- Select Element
- Create Global Web Page
- Create Control from Web Controls Tab
- Highlight Control
- Analyze the Object Explorer Hierarchy
- Investigate the difference between IE6 and IE7

Exploring the Interrogation Function

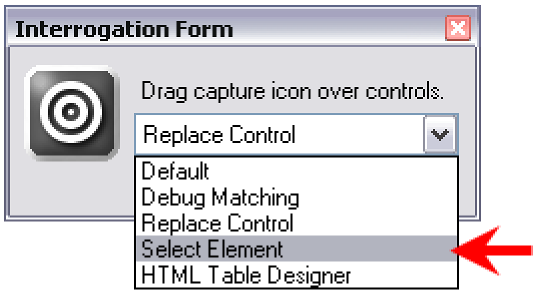
The basic interrogation function for web applications involves dragging the target icon over a target control. In many cases, this simple method for choosing targets enables you to identify most, if not all, of the targets you require.

In some instances, you will need to use additional functions to identify targets:

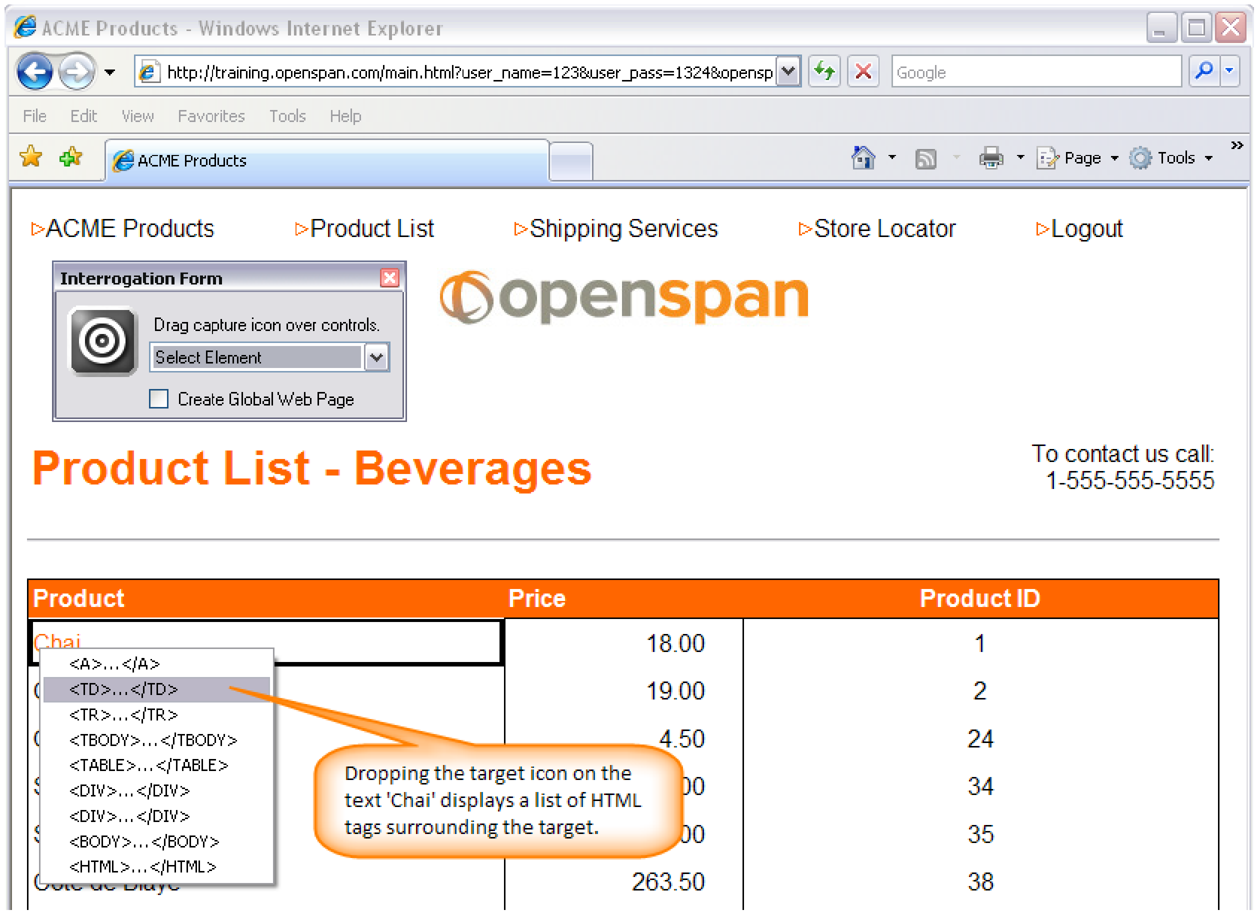
- Select Element
- Create Global Web Page
- Create Control from Web Controls tab

Select Element

Dragging the Interrogator target (bulls-eye) icon over a target on a web page may not always provide enough precision to highlight the target that you want to add to the solution. When this occurs, use the **Select Element** option.



The **Select Element** option displays a list of HTML tags surrounding the target you select on the HTML page. You can then select the target based on the HTML tag surrounding the target. In the following image, the **Chai** text in the Product column is interrogated.



The selection options for this target are:

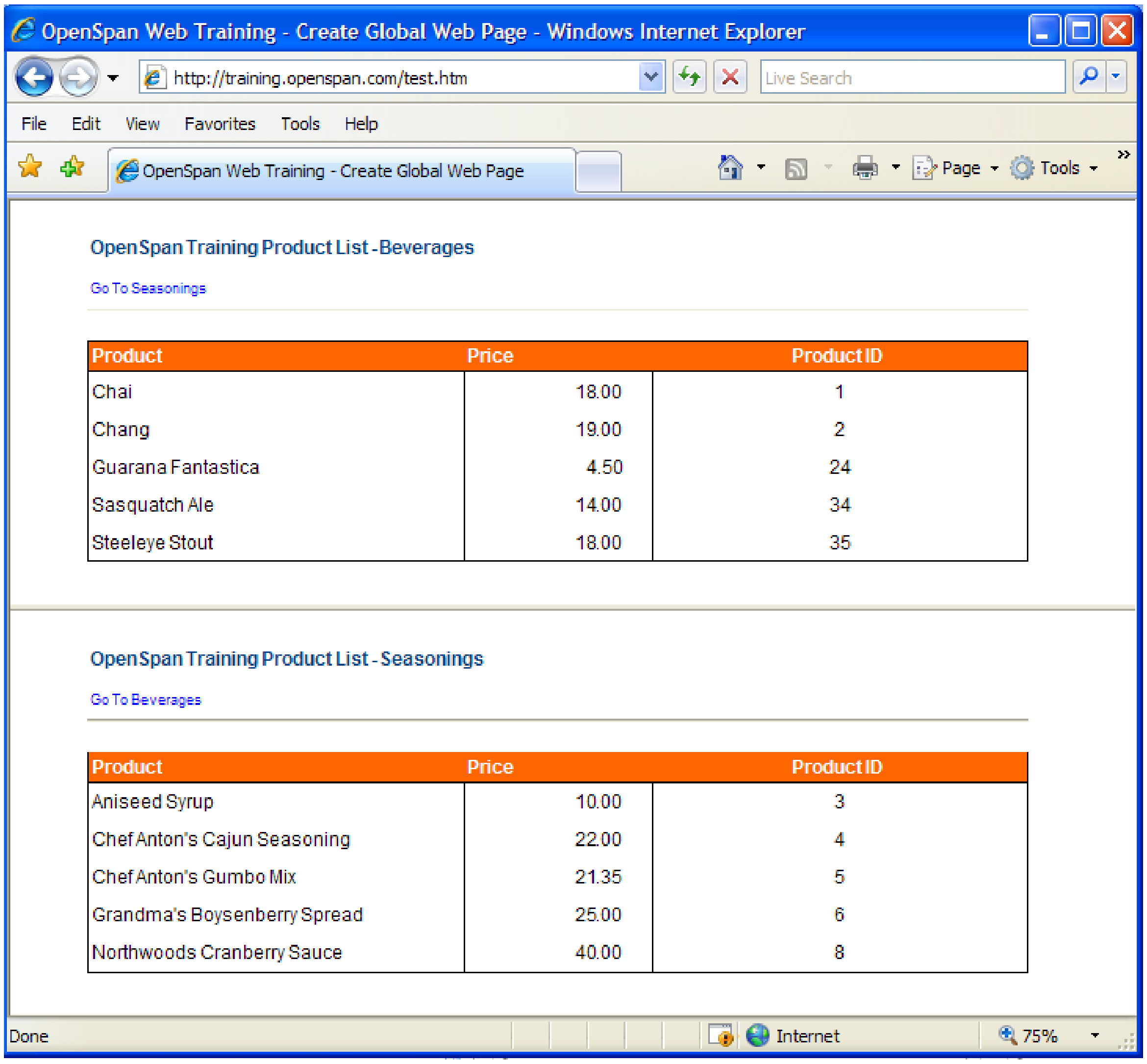
A	Anchor
TD	Table data cell (row 1, column 0)
TR	Table row
TBODY	Table Body
TABLE	HTML Table
DIV	All content contained within the Div style container
DIV	All content contained within the Div style container
BODY	Document body
HTML	Document root element

Create Global Web Page

When interrogating web applications that contain multiple frames, you can use the **Create Global Web Page** option to assign the interrogated objects to an independent global page. This is required when objects in a frame can exist on various pages within the web site, depending on the user's interaction with the site. This feature should also be used when there are different versions of Internet Explorer in your environment.

Example Global Web Page:

The following example web application contains a frameset with two frames. Each frame displays a separate web page. The top frame displays the Bev_List.html page and the bottom frame displays the Season_List.html page from the OpenSpan training website.


OpenSpan Web Training - Create Global Web Page - Windows Internet Explorer
http://training.openspan.com/test.htm
Live Search
File Edit View Favorites Tools Help
OpenSpan Web Training - Create Global Web Page
Page Tools
Done
Internet
75%

OpenSpan Training Product List - Beverages

Go To Seasonings

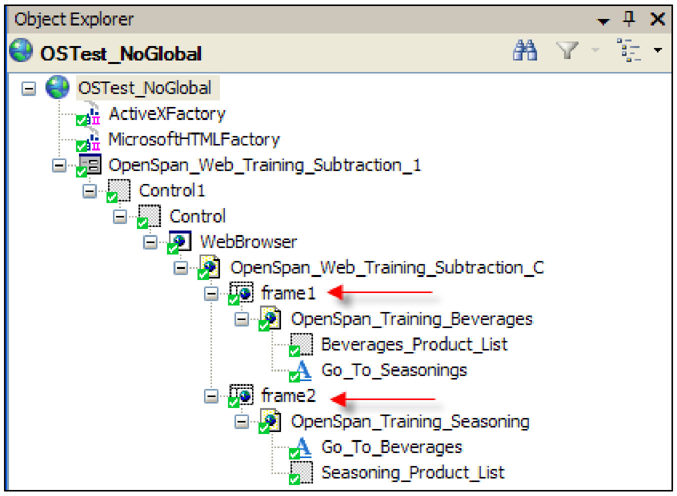
Product	Price	Product ID
Chai	18.00	1
Chang	19.00	2
Guarana Fantastica	4.50	24
Sasquatch Ale	14.00	34
Steeleye Stout	18.00	35

OpenSpan Training Product List - Seasonings

Go To Beverages

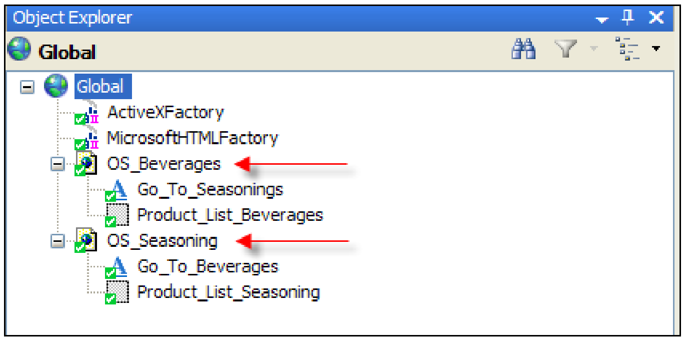
Product	Price	Product ID
Aniseed Syrup	10.00	3
Chef Anton's Cajun Seasoning	22.00	4
Chef Anton's Gumbo Mix	21.35	5
Grandma's Boysenberry Spread	25.00	6
Northwoods Cranberry Sauce	40.00	8

Interrogating these frames, without using **Create Global Web Page,** yields separate frame objects under a common page in the Object Explorer as shown in the following hierarchy: (This example used Internet Explorer version 7.)



```
OSTest_NoGlobal
    ActiveXFactory
    MicrosoftHTMLFactory
    OpenSpan_Web_Training_Subtraction_1
        Control1
            Control
                WebBrowser
                    OpenSpan_Web_Training_Subtraction_C
                        frame1
                            OpenSpan_Training_Beverages
                                Beverages_Product_List
                                Go_To_Seasonings
                        frame2
                            OpenSpan_Training_Seasoning
                                Go_To_Beverages
                                Seasoning_Product_List
```

However, if the frames are interrogated using **Create Global Web Page**, the targets are returned to the Object Explorer as separate page objects as shown below:



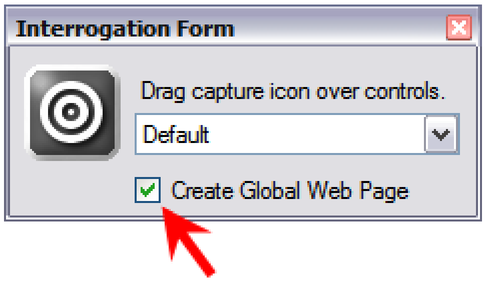
```
Global
    ActiveXFactory
    MicrosoftHTMLFactory
    OS_Beverages
        Go_To_Seasonings
        Product_List_Beverages
    OS_Seasoning
        Go_To_Beverages
        Product_List_Seasoning
```

Note: When using **Create Global Web Page**, OpenSpan does not create controls for any of the web page parent objects (e.g., Browser, Tabs, or IE Window). This enables OpenSpan to match the interrogated web pages in any of the supported versions of Internet Explorer.

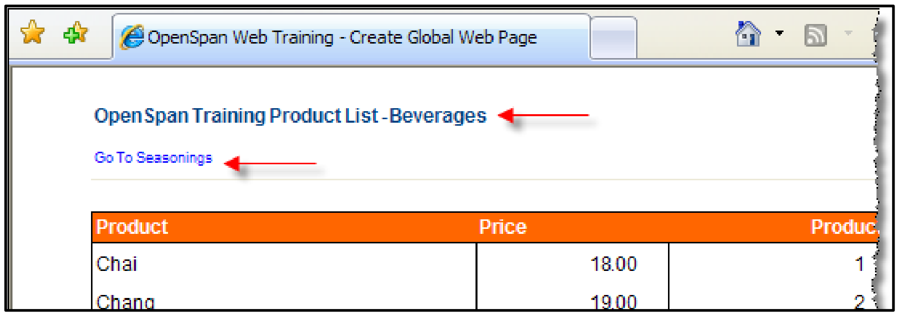
Exercise – Global Web Page

This exercise enables you to use the OpenSpan training web site, http://training.openspan.com/test.htm to practice interrogating a web application using the **Create Global Web** Page option.

1. In OpenSpan Studio, create a new project named **Global Web Page**.
2. Add a Web application to the project and name it **Training Global Web Page**.
3. Set the **StartPage** for the Web application to: **http://training.openspan.com/test.htm**
4. Select the **Start Interrogation** button.
5. On the **Interrogation Form** dialog, select the **Create Global Web Page** check box.

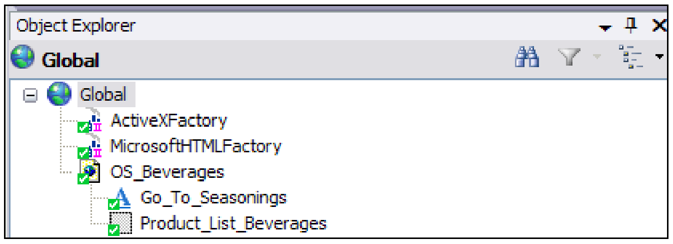


6. Interrogate the **Go To Seasonings** link and the **OpenSpan Training Product List – Beverages** heading in the top frame.

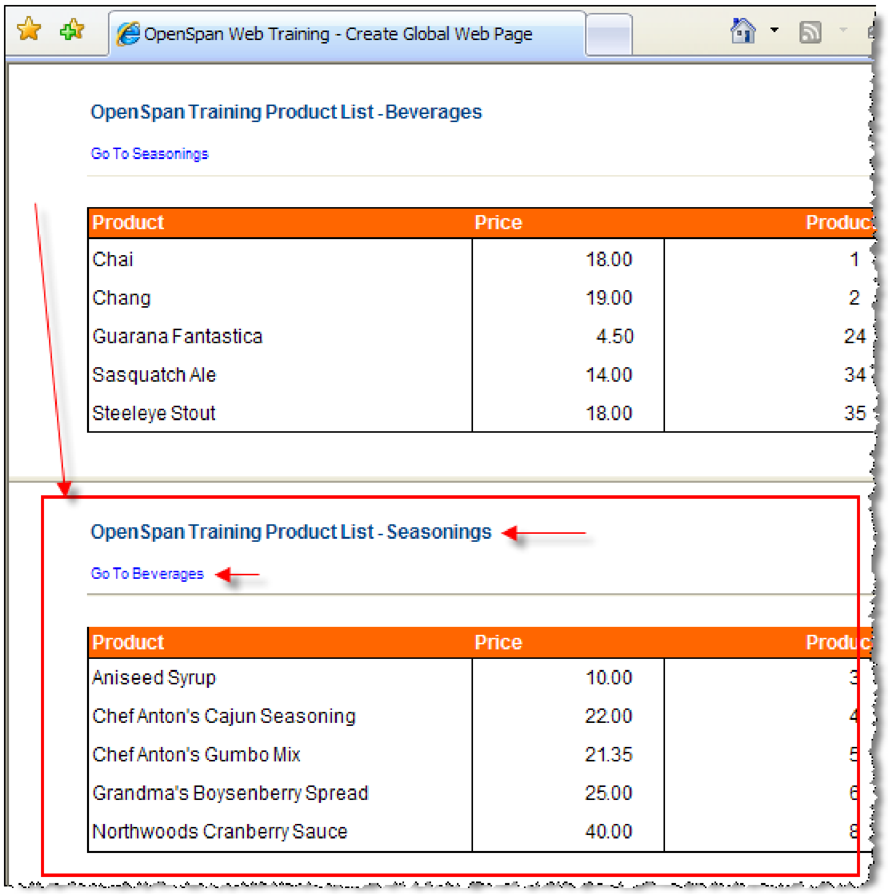

OpenSpan Web Training - Create Global Web Page
OpenSpan Training Product List -Beverages
Go To Seasonings

Product	Price	Produc
Chai	18.00	1
Chang	19.00	2

7. Rename the Page control to **OS_Beverages** and the Web control for the heading to **Product_List_Beverages**. An example of the Object Explorer showing the resulting controls follows:



8. Interrogate the **Go To Beverages** link and the **OpenSpan Training Product List – Seasonings** heading in the bottom frame.



OpenSpan Training Product List - Beverages

Go To Seasonings

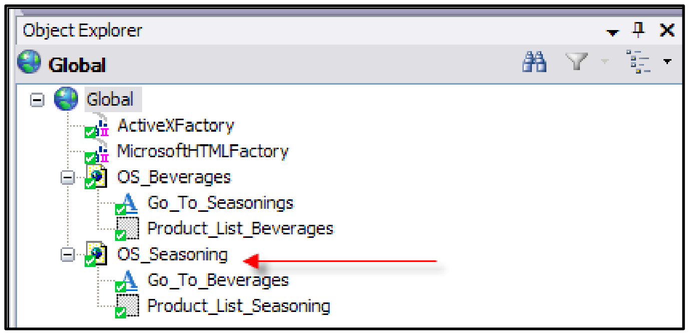
Product	Price	Produc
Chai	18.00	1
Chang	19.00	2
Guarana Fantastica	4.50	24
Sasquatch Ale	14.00	34
Steeleye Stout	18.00	35

OpenSpan Training Product List - Seasonings

Go To Beverages

Product	Price	Produc
Aniseed Syrup	10.00	3
Chef Anton's Cajun Seasoning	22.00	4
Chef Anton's Gumbo Mix	21.35	5
Grandma's Boysenberry Spread	25.00	6
Northwoods Cranberry Sauce	40.00	8

9. Rename the Page control to **OS_Seasoning** and the Web control for the heading to **Product_List_Seasoning**. An example of the Object Explorer showing the resulting controls follows:



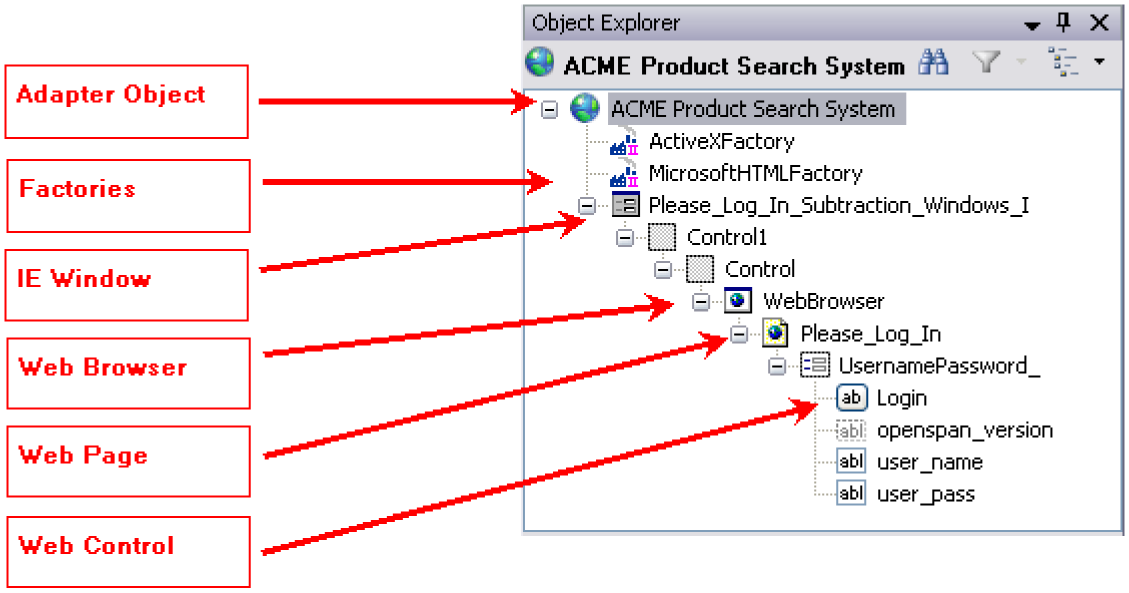
```
Global
    ActiveXFactory
    MicrosoftHTMLFactory
    OS_Beverages
        Go_To_Seasonings
        Product_List_Beverages
    OS_Seasoning
        Go_To_Beverages
        Product_List_Seasoning
```

10. With the Interrogator running, change the Address to http://training.openspan.com/Season_list.html. The browser replaces the test.htm page with the Season_list page. Note that the controls interrogated when the Season_List page displayed in the bottom frame of the Test.htm page remain matched.

11. Stop the interrogation and save the solution.

Interrogated Objects

As you retrieve target objects into OpenSpan Studio, the targets display in the Object Explorer. The Object Explorer is a hierarchy that shows how the interrogated objects are related through their parent objects.--xsos



ACME Product Search System
ActiveXFactory
MicrosoftHTMLFactory
Please_Log_In_Subtraction_Windows_I
Control1
Control
WebBrowser
Please_Log_In
UsernamePassword_
Login
openspan_version
user_name
user_pass

Parts of the Object Explorer Hierarchy

The top (or root) node of the hierarchy is the adapter object – Web adapter. This object contains the functionality required to integrate OpenSpan Studio with a web application.

Beneath the adapter object are the Factory objects. The Factories enable OpenSpan to work with specific platforms. Also beneath the adapter object are the windows that contain the IE Browser. For example, the window name is **Please_Log_In_Subtraction_Windows_I**. Directly beneath this is Control1 which is the tab control for IE7 (this object is not present in IE6). Beneath this is the Shell DocObject which contains the browser.

The next node in the Object Explorer is the Web Browser. Following the Web Browser object, OpenSpan creates page objects for each page you interrogate after the **StartPage**. The controls that you interrogate on each page are included in the hierarchy as child objects under the appropriate pages.

The table below lists the types of objects OpenSpan Studio includes in the hierarchy:

Web Objects	
Anchor	Input = Radio
Form	Input = Submit
Frame	Input = Text
IEFrame	Object (Generic embedded object)
Image	Select

Web Objects	
Image Button	Table
Input = Button	Table data cell (TD)
Input = CheckBox	Table Row (TR)
Input = Hidden	TextArea
Input = Password	Webpage

Create Control for Hidden Target

You can include hidden web targets in solutions. For example, suppose there is a hidden label that contains information required for your solution. Since you cannot see the label, you cannot drop the target icon on it.

Use the **Web Controls** tab to locate the hidden object and add it to your solution. The **Web Controls** tab also provides a convenient way to list all web targets from a page that is currently open in the interrogator. Later in this chapter, you will get a chance to interrogate a hidden web control.

Exercise - Interrogating Web Applications

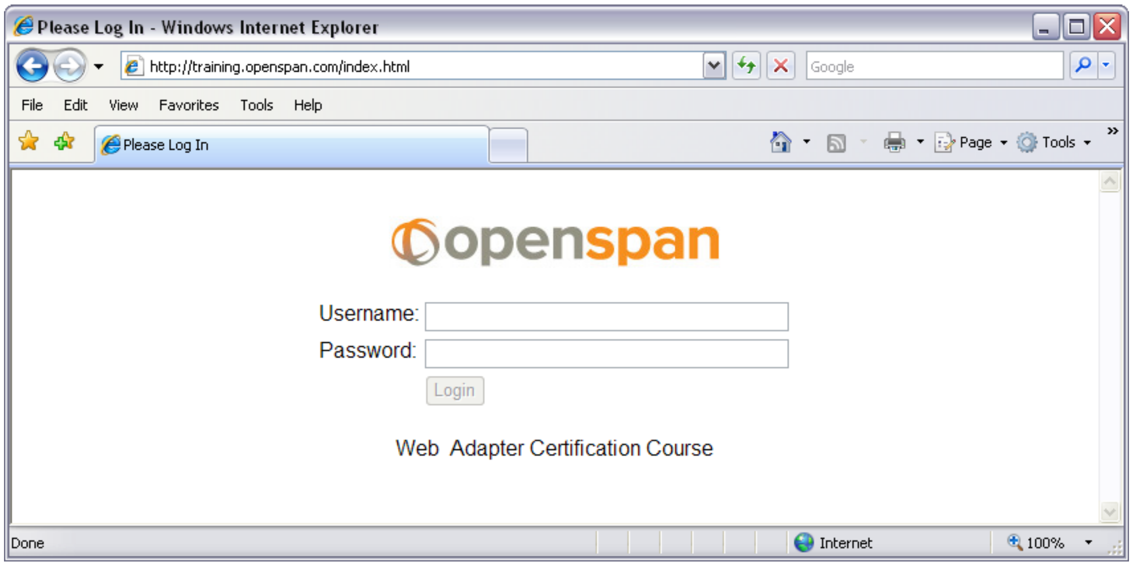
This section of the course gives you hands-on experience with the following interrogation techniques and functions:

- **Using Target Icon**
  Concepts: *Interrogating basic objects*
- **Using Select Element**
  Concepts: *Using the Select Element option to add objects to a solution*
- **Creating a Control**
  Concepts: *Using the Web tab to locate a target and create the control*

1. In OpenSpan Studio, create a new solution named **Web Adapter Certification**.
2. Add a Web Application project item named **ACME Product Search System**.
3. Set the **StartPage** property for the Web application to:

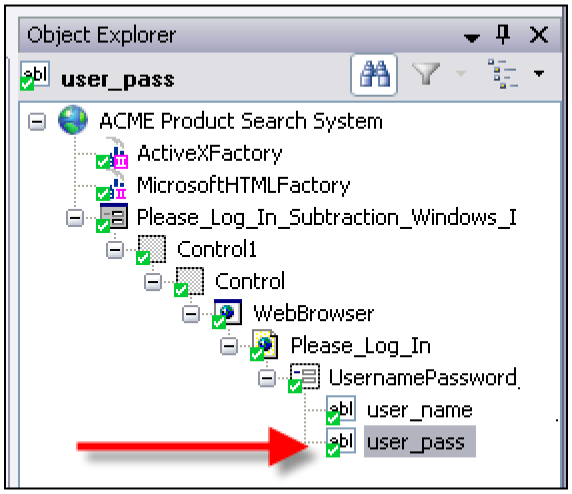
   http://training.openspan.com/index.html.
4. On the **ACME Products Search System** project item, select the **Start Interrogation** button.

   The **Login** page for the OpenSpan Web Adapter Certification Course should open in your browser.


Please Log In - Windows Internet Explorer
http://training.openspan.com/index.html
Google
File Edit View Favorites Tools Help
Please Log In
Page Tools
openspan
Username:
Password:
Login
Web Adapter Certification Course
Done
Internet
100%

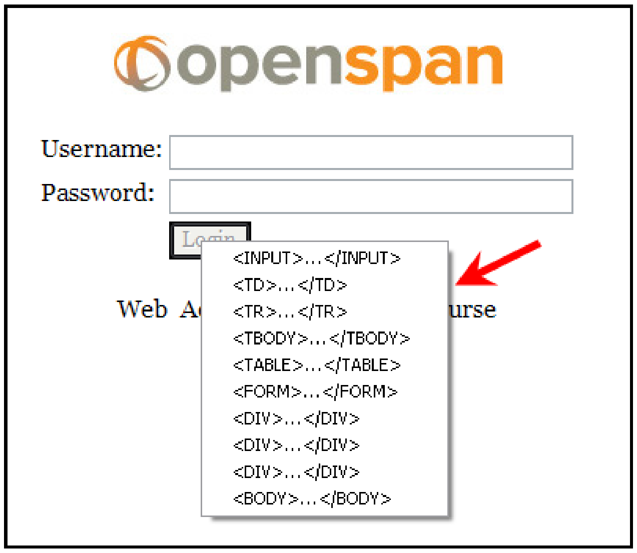
5. With the **Interrogation Form** dialog option set to **Default,** drag the target icon over the **Username** and **Password** textboxes. For this exercise, **do not** select the **Create Global Web Page** option.

Your Object Explorer should look like the following:

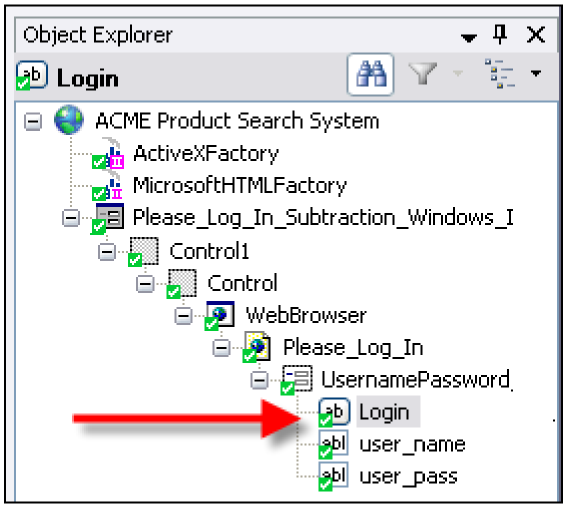


Select Element

6. Change the **Interrogation Form** dialog option to **Select Element** and then drag the target icon over the **Login** button. When you release the mouse button a list of HTML tags display. These are the tags surrounding the Button (Submit) object.

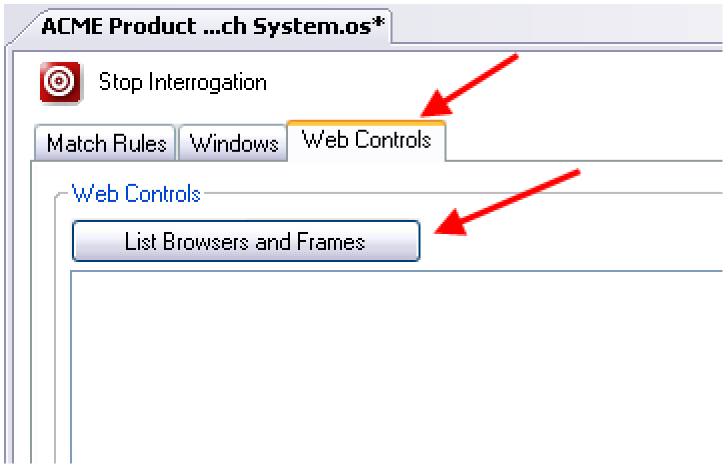


7. Select the **Input** tag. This is the HTML tag that represents the Submit button object. Once you select the **Input** tag, the object is added to the Object Explorer.

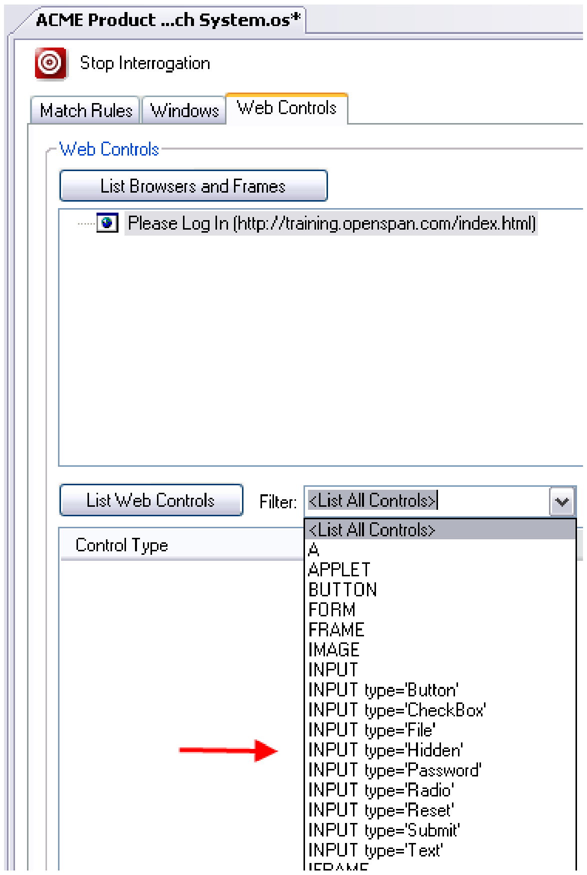


Create Hidden Control

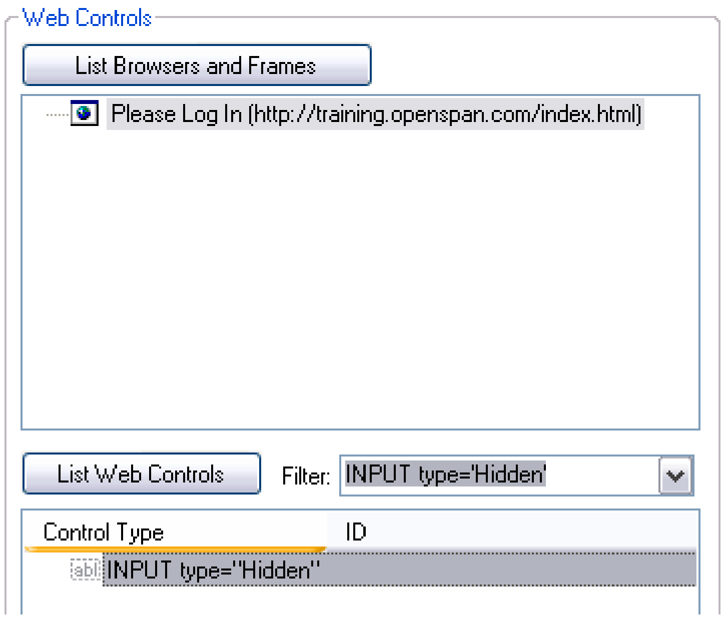
8. Open the **Web Controls** tab in the design window. Click the **List Browsers and Frames** button to display the **Please Log In** page.



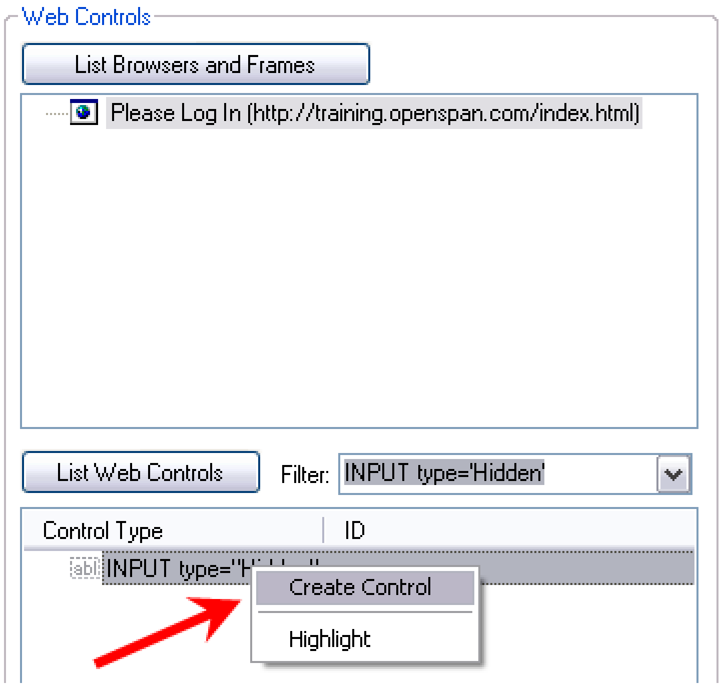
9. In the **Filter** drop-down list, select **INPUT type = 'Hidden'**.



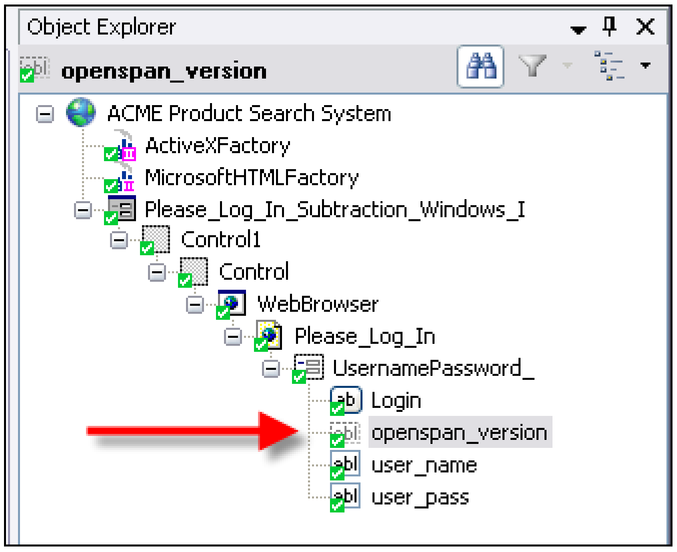
10. Click the **List Web Controls** button. A hidden textbox control type displays.



11. Right-click the **Input Hidden Control** and select **Create Control** from the context menu.



12. The hidden textbox is added to the Object Explorer and displays as matched.



13. Stop the interrogation, save the solution, and close the **ACME Products Search System.os** design tab.

This page intentionally left blank.

CHAPTER 3: UNDERSTANDING WEB ADAPTER MATCH RULES

This chapter describes and presents examples of the matching process for web applications and controls. Match Rules for the following object types are covered:

- Anchor
- Form
- Image
- Input = Submit
- Input = Text
- Generic Web Object
- Select
- Table
- Table Data Cell (TD)
- Table Row (TR)
- Webpage

Web Adapter Match Rules

OpenSpan Studio uses Match Rules to uniquely identify the objects in the web application that you want to use in your solution. When you interrogate a web application, OpenSpan Studio identifies the objects through a sequence of Match Rules. The Match Rules are logical statements related to the properties of the objects.

For web applications, OpenSpan reads the surrounding HTML for the targets on the selected page and creates a table linking the objects to the HTML tags. Through use of the tags, the objects are uniquely identified. The Match Rule (or Rules) used to match a particular object depends on the object type. For example, a link or Anchor element can be matched using the following anchor attribute match rules:

- Anchor Name
- Anchor Target
- Anchor URL

As well as default web control match rules:

- Control Children
- Element ID
- Element Index
- Element Inner Text
- Element Path
- Table Section Row
- Attribute Value

Web Application Match Rules

Now that you have seen the Match Rules in action, let's take a look at working with Match Rules within the OpenSpan Studio designer. The first few objects that you create when interrogating a web application without the **Create Global Web Page** selected are actually windows objects.

- IE Frame
- Shell Doc View
- Browser

The match rules applied to these objects are Windows based. Use the following steps to explore the Windows Match Rules used for the Web application.

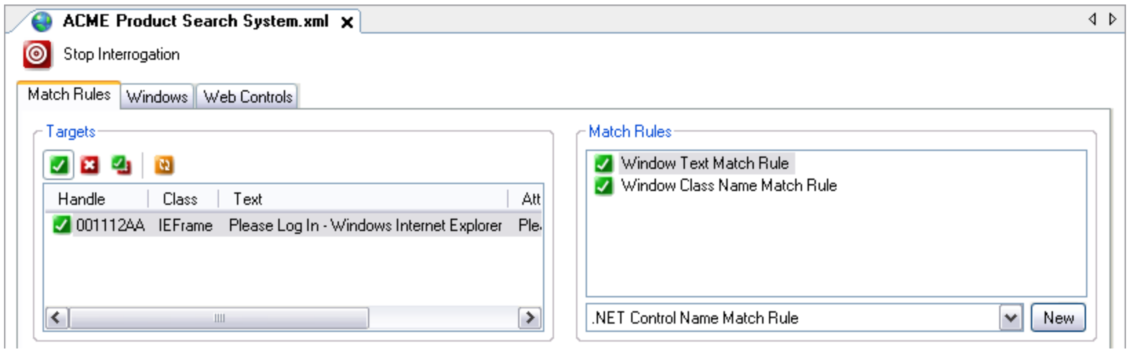
ActiveX Factories – Module Name Match Rule

The default match rule for ActiveX is the Module Name match rule. No editing is usually required for this match rule. Other available match rules for ActiveX are:

- Control Children Match Rule
- Module Version Match Rule

IE Frame - Window Text and Class Name Match Rules

As you interrogate the web application, OpenSpan matches the **IEFrame** Windows form, which parents the browser and web application objects.



Handle	Class	Text	Att
001112AA	IEFrame	Please Log In - Windows Internet Explorer	Ple.

- ✓ Window Text Match Rule
- ✓ Window Class Name Match Rule

.NET Control Name Match Rule

The default match rules applied for the **IEFrame** are:

- Window Text
- Window Class

These are default match rules for WinForm objects.

Window Text Match Rule

The Window Text Match Rule is the default match rule used to match window objects – windows, MDI child windows, window control labels, etc. The main properties of this match rule are:

- **Culture:** Language (culture) used by target application end users. Default **(User Culture)** uses the operating system setting (Control Panel | Regional and Language Options).
- **IgnoreCase:** Boolean, default **True**, which indicates that text can appear in either upper or lower case.
- **Mode:** Default **Simple** – text matched literally. Mode types available include *Regex*, *StartsWith*, *EndsWith*, etc. **Regex** mode indicates that the Text property (below) contains Regex syntax.
- **Text:** Target object label text, such as window title bar text or a control button label. For the sample CRM.exe application, the Login window text is *Login*. Modify text to support **Mode** selection.

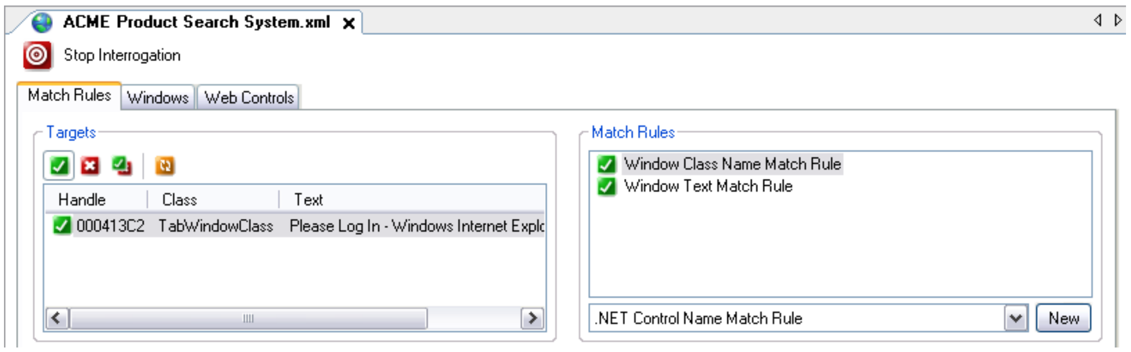
Window Class Name Match Rule

The **Windows Class Name Match Rule** is the default match rule used to match Microsoft Windows application objects such as textboxes, buttons etc. The main property for this match rule is:

- **ClassName**: Gets or sets the Class Name for the Match Rule. Example Class Name for iexplore.exe is *IEFrame*.

TabWindowClass - Window Text and Class Name Match Rules

OpenSpan uses **Window Class Name Match Rule** and the **Window Text Match Rule** to match the control under the IEFrame for Internet Explorer version 7.



Handle	Class	Text
000413C2	TabWindowClass	Please Log In - Windows Internet Explo

Window Class Name Match Rule
Window Text Match Rule

.NET Control Name Match Rule

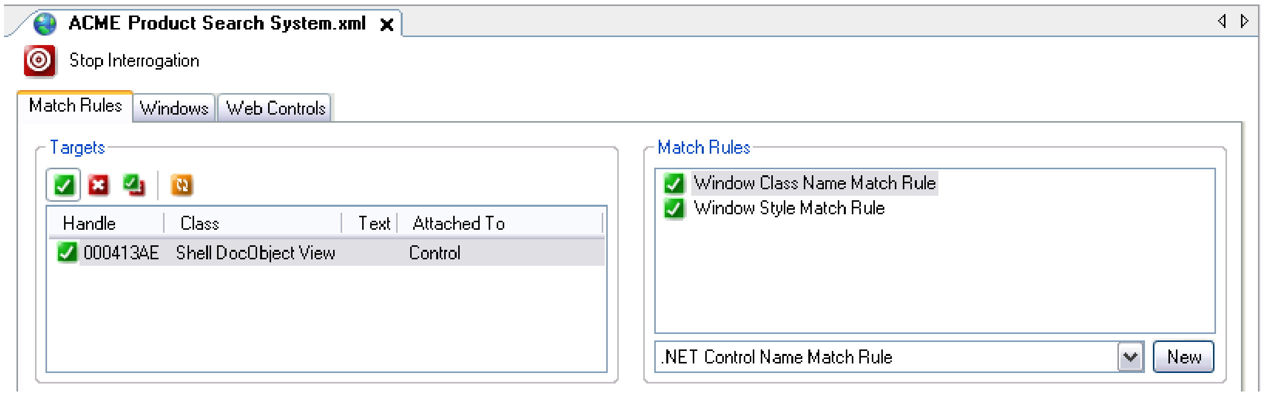
The default match rules applied for the **TabWindowClass** are:

- Window Text
- Window Class

These are default match rules for TabWindowClass objects.

Shell DocObject View - Window Style Match Rule

OpenSpan uses the **Window Class Name** and **Window Style Match Rules** to match the control under the **IEFrame** for Internet Explorer version 6 (for Internet Explorer version 7 you will see a TabWindowClass control).

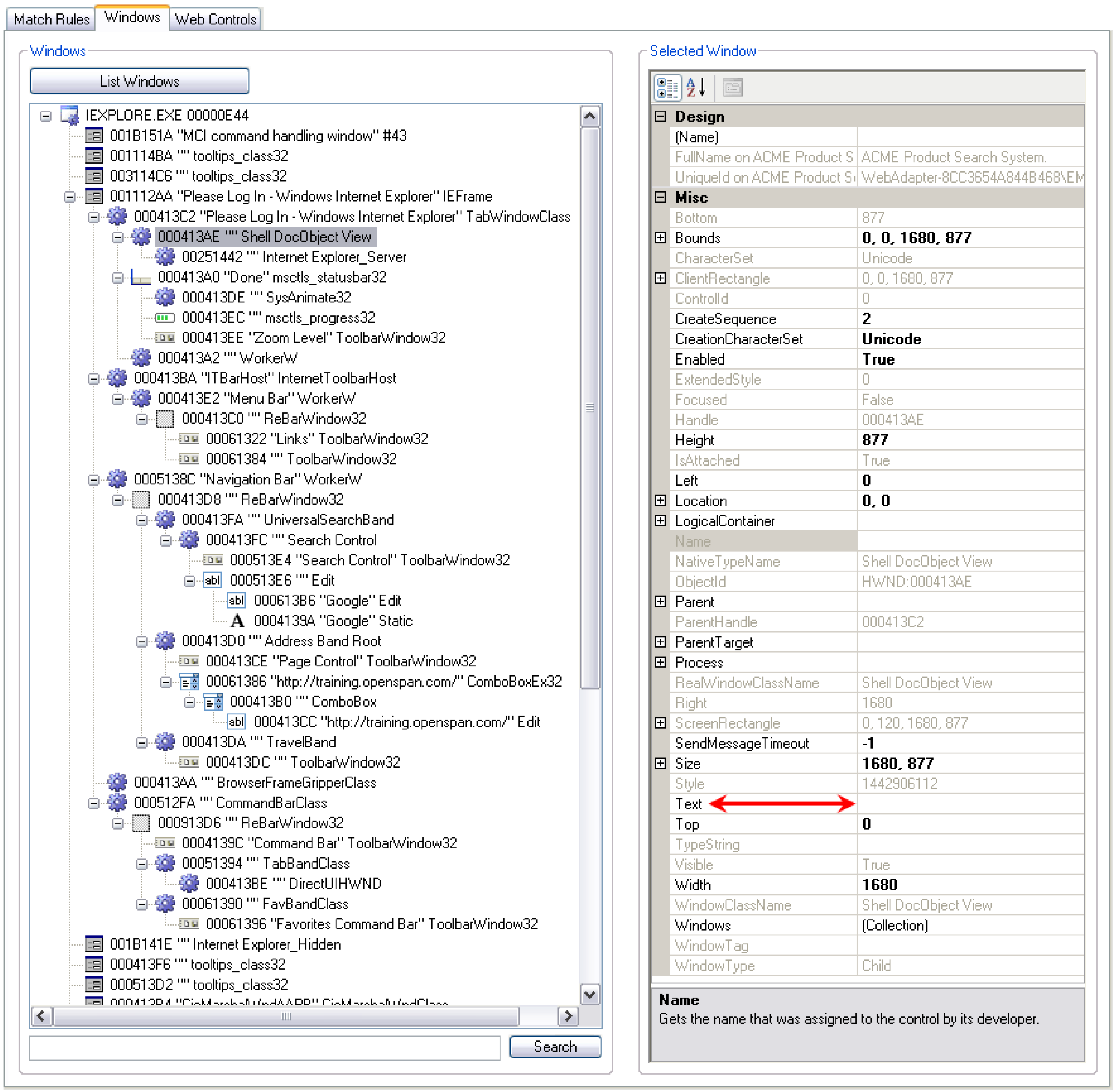


Handle	Class	Text	Attached To
✓ 000413AE	Shell DocObject View		Control

Match Rules:
- ✓ Window Class Name Match Rule
- ✓ Window Style Match Rule

.NET Control Name Match Rule

In this case, the **Window Text Match Rule** could not be applied since there is no text associated with the target. You verify this by opening the **Windows** tab and locating the **Shell DocObject View** in the **Windows** group box.



```
IEXPLORE.EXE 00000E44
    001B151A "MCI command handling window" #43
    001114BA "" tooltips_class32
    003114C6 "" tooltips_class32
    001112AA "Please Log In - Windows Internet Explorer" IEFrame
        000413C2 "Please Log In - Windows Internet Explorer" TabWindowClass
            000413AE "" Shell DocObject View
                00251442 "" Internet Explorer_Server
            000413A0 "Done" msctls_statusbar32
                000413DE "" SysAnimate32
                000413EC "" msctls_progress32
                000413EE "Zoom Level" ToolbarWindow32
            000413A2 "" WorkerW
        000413BA "ITBarHost" InternetToolbarHost
            000413E2 "Menu Bar" WorkerW
                000413C0 "" ReBarWindow32
                    00061322 "Links" ToolbarWindow32
                    00061384 "" ToolbarWindow32
        0005138C "Navigation Bar" WorkerW
            000413D8 "" ReBarWindow32
                000413FA "" UniversalSearchBand
                    000413FC "" Search Control
                        000513E4 "Search Control" ToolbarWindow32
                        000513E6 "" Edit
                            000613B6 "Google" Edit
                            0004139A "Google" Static
                000413D0 "" Address Band Root
                    000413CE "Page Control" ToolbarWindow32
                    00061386 "http://training.openspan.com/" ComboBoxEx32
                        000413B0 "" ComboBox
                            000413CC "http://training.openspan.com/" Edit
                000413DA "" TravelBand
                    000413DC "" ToolbarWindow32
        000413AA "" BrowserFrameGripperClass
        000512FA "" CommandBarClass
            000913D6 "" ReBarWindow32
                0004139C "Command Bar" ToolbarWindow32
                00051394 "" TabBandClass
                    000413BE "" DirectUIHWND
                00061390 "" FavBandClass
                    00061396 "Favorites Command Bar" ToolbarWindow32
    001B141E "" Internet Explorer_Hidden
    000413F6 "" tooltips_class32
    000513D2 "" tooltips_class32
```

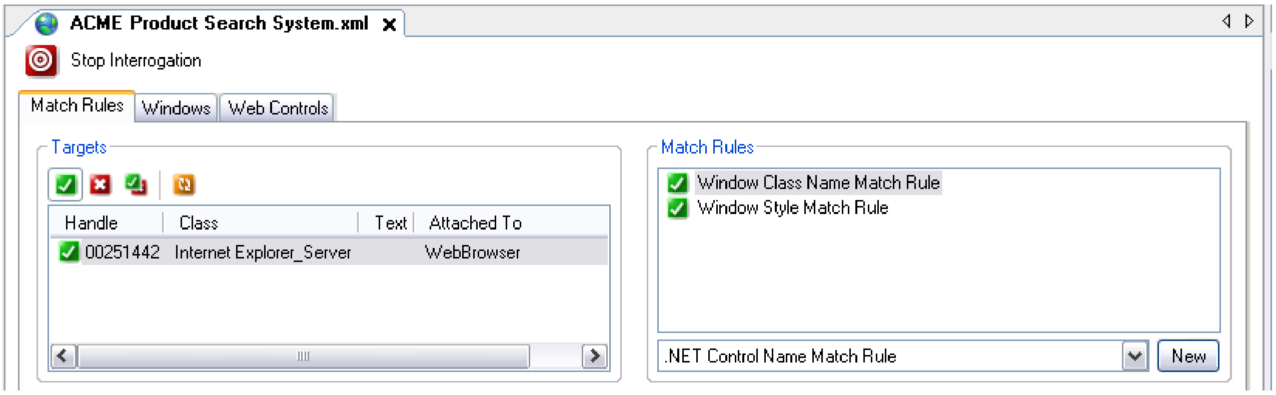
Design	
(Name)	
FullName on ACME Product S	ACME Product Search System.
UniqueId on ACME Product S	WebAdapter-8CC3654A844B468\EM
Misc	
Bottom	877
Bounds	**0, 0, 1680, 877**
CharacterSet	Unicode
ClientRectangle	0, 0, 1680, 877
ControlId	0
CreateSequence	**2**
CreationCharacterSet	**Unicode**
Enabled	**True**
ExtendedStyle	0
Focused	False
Handle	000413AE
Height	**877**
IsAttached	True
Left	**0**
Location	**0, 0**
LogicalContainer	
Name	
NativeTypeName	Shell DocObject View
ObjectId	HWND:000413AE
Parent	
ParentHandle	000413C2
ParentTarget	
Process	
RealWindowClassName	Shell DocObject View
Right	1680
ScreenRectangle	0, 120, 1680, 877
SendMessageTimeout	**-1**
Size	**1680, 877**
Style	1442906112
Text	
Top	**0**
TypeString	
Visible	True
Width	**1680**
WindowClassName	Shell DocObject View
Windows	(Collection)
WindowTag	
WindowType	Child

Name
Gets the name that was assigned to the control by its developer.

To uniquely match this window control, OpenSpan Studio uses the **Window Style Match Rule** and the **Window Class Match Rule**. The style for the Shell DocObject View is unique.

Browser – Window Class and Window Style Match Rules

OpenSpan Studio uses the same match rules – Window Class and Window Style to match the WebBrowser object:



Handle	Class	Text	Attached To
00251442	Internet Explorer_Server		WebBrowser

- Window Class Name Match Rule
- Window Style Match Rule

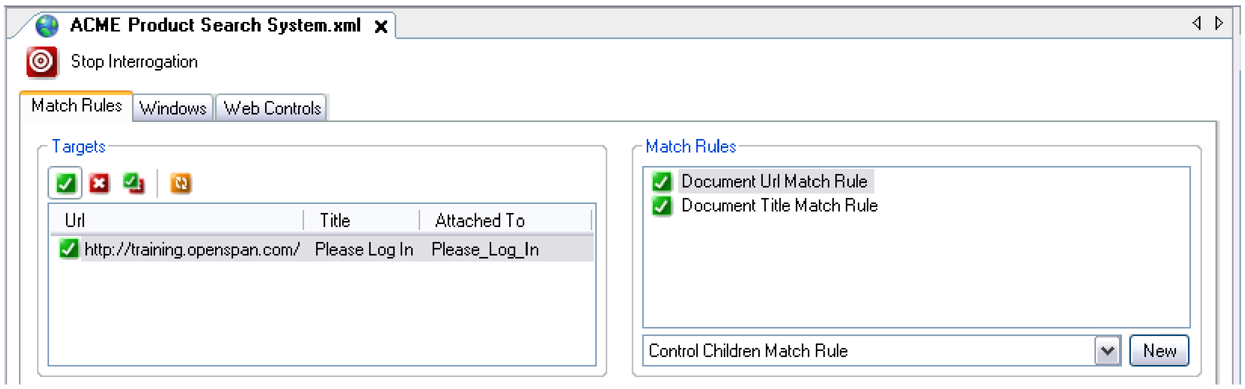
HTML Page - Web Object Matching

The first web object created in the Object Explorer is the Web Page. OpenSpan Studio uses the following Match Rules to identify the page.

- Document URL
- Document Title

For example, with the Web Adapter Certification site, the main page – **Please_Log_In**, is matched as follows.

- Document URL – URL for site
- Document Title – OpenSpan Web Adapter Certification Course



Url	Title	Attached To
http://training.openspan.com/	Please Log In	Please_Log_In

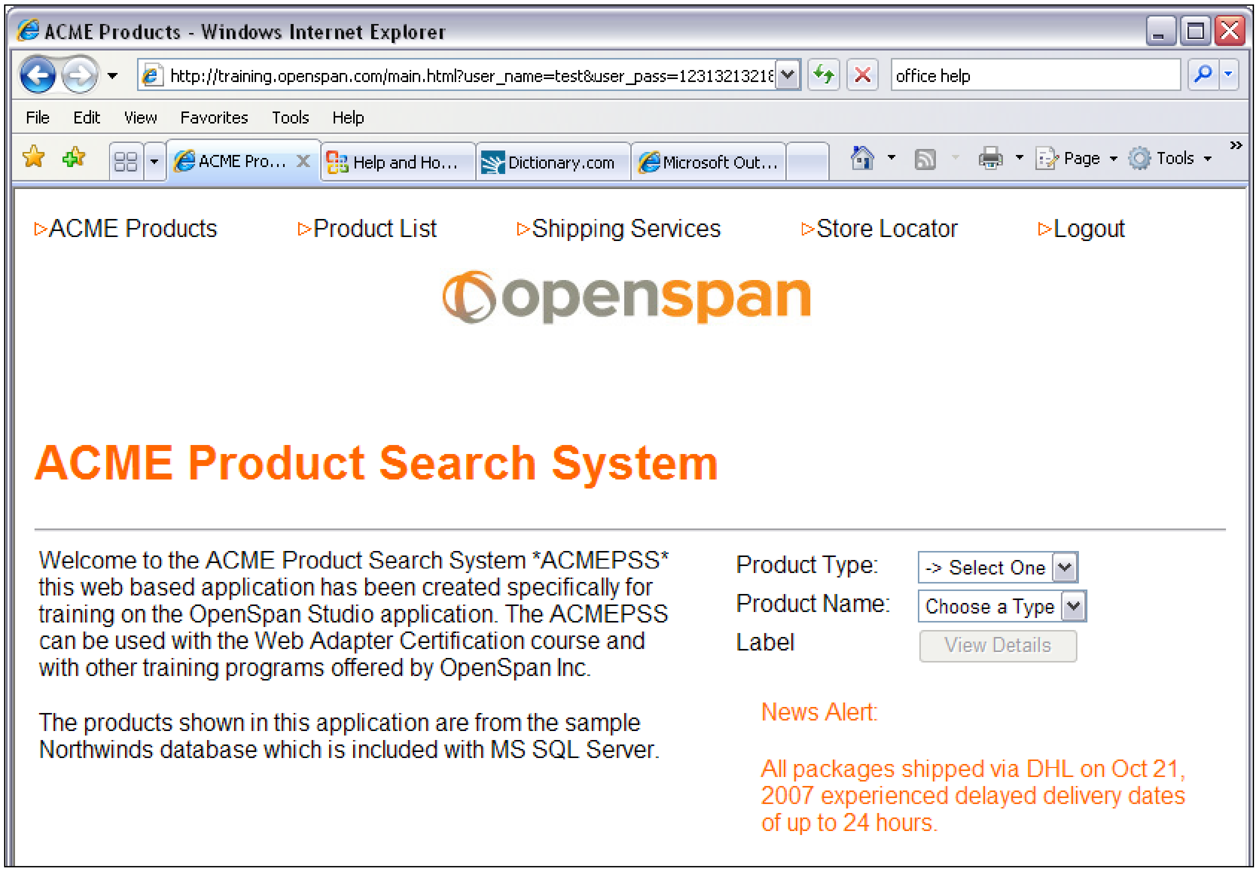
- Document Url Match Rule
- Document Title Match Rule

Default Match Rules

As you interrogate the objects within the application, OpenSpan uses the match rules best suited to uniquely identify the objects. There are special match rules for the various web control types. However, there are default match rules that can be applied to any web control.

- Attribute Value
- Control Children
- Element ID
- Element Index
- Element Inner Text
- Element Path

OpenSpan Studio may use one or more of the above match rules to uniquely match a target. For example, in the ACME Product Search System application, OpenSpan Studio uses the **Element Path Match Rule** to match the Heading text on the Home page.



ACME Product Search System

Welcome to the ACME Product Search System *ACMEPSS* this web based application has been created specifically for training on the OpenSpan Studio application. The ACMEPSS can be used with the Web Adapter Certification course and with other training programs offered by OpenSpan Inc.

The products shown in this application are from the sample Northwinds database which is included with MS SQL Server.

Product Type: -> Select One

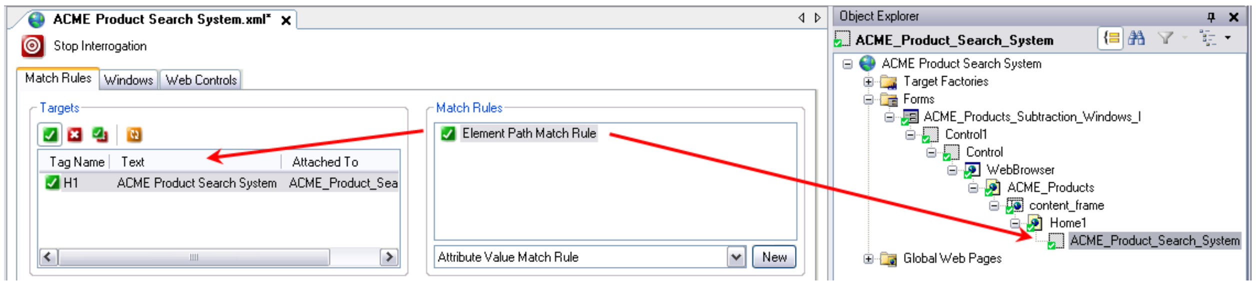
Product Name: Choose a Type

Label

View Details

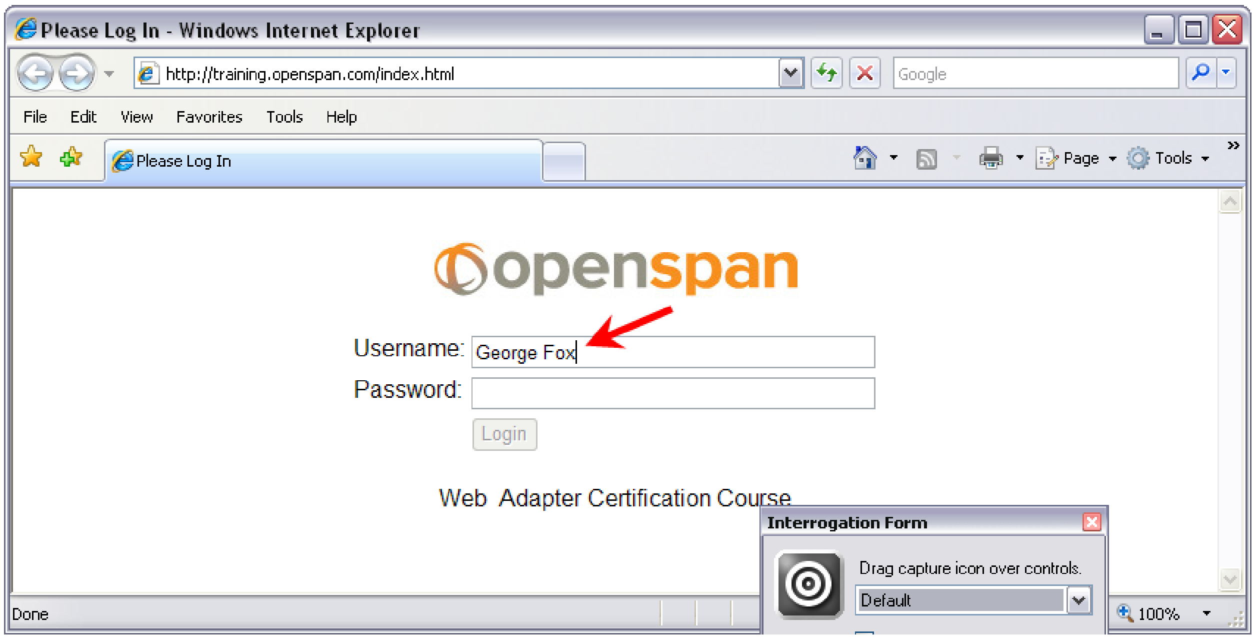
News Alert:

All packages shipped via DHL on Oct 21, 2007 experienced delayed delivery dates of up to 24 hours.

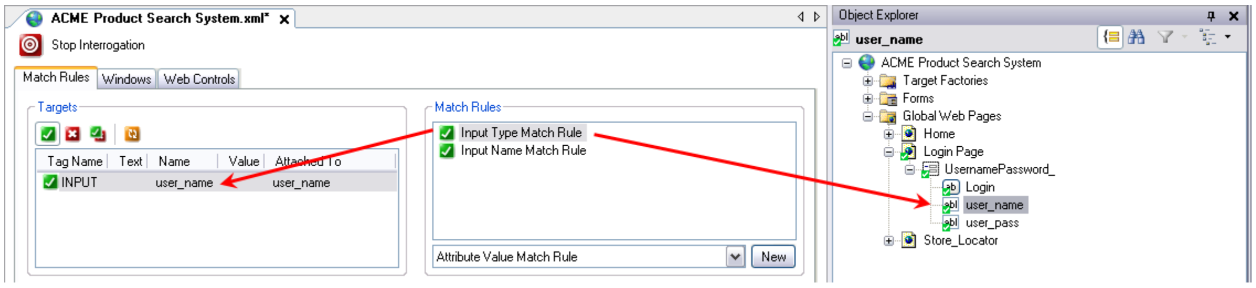


Tag Name	Text	Attached To
H1	ACME Product Search System	ACME_Product_Sea

The **Element Path Match Rule** is used since there are no attribute match rules that apply to this object. In contrast, when matching a standard web control, OpenSpan Studio uses the match rules developed specifically for the control. For example, the **Username** on the Login page is matched with the **Input Type** and **Input Name Match Rules**.


Please Log In - Windows Internet Explorer
http://training.openspan.com/index.html
Google
File
Edit
View
Favorites
Tools
Help
Please Log In
Page
Tools
openspan
Username:
Password:
Login
Web Adapter Certification Course
Interrogation Form
Drag capture icon over controls.
Default
Done
100%

Username: George Fox



Tag Name	Text	Name	Value	Attached To
INPUT		user_name		user_name

The following shows the default match rules used for standard web controls:

Anchor ▪ Anchor Target ▪ Anchor URL	**Generic Web Control** ▪ Object Class ID ▪ Object Code Type ▪ Object Code ▪ Object Name ▪ Object Type
Form ▪ Form Method ▪ Form Name ▪ Form Target ▪ Frame Name ▪ Frame Source	**Select** ▪ Select Index ▪ Select Is Multiple ▪ Select Name ▪ Select Option Count ▪ Select Size
Image ▪ Image Alternate Text ▪ Image Name ▪ Image Source	**Table** ▪ Table Border ▪ Table Height ▪ Table Width
Input ▪ Image Button Alternate Text ▪ Image Button Source ▪ Input Index ▪ Input Name ▪ Input Size ▪ Input Type ▪ Input Text	**Table Data Cell (TD)** ▪ Table Cell Column Index ▪ Table Cell Height ▪ Table Cell Padding ▪ Table Cell Row ▪ Table Cell Row Index ▪ Table Cell Spacing ▪ Table Cell Width

Exercise - Attribute Value Match Rule

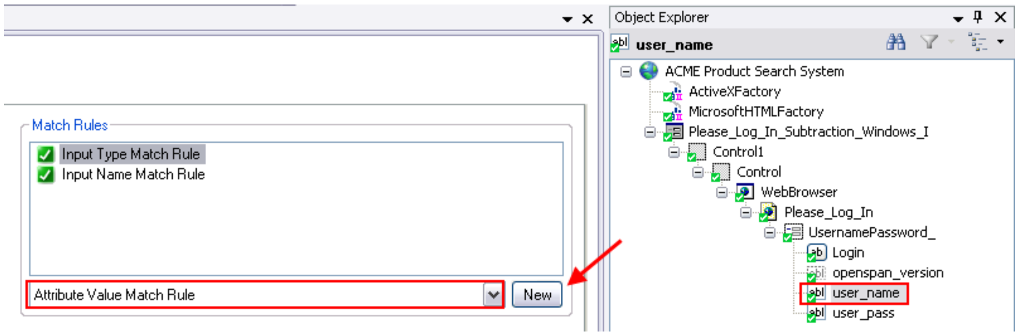
To use the **Attribute Value Match Rule**, you select the attribute and set the value to match. Use the following steps to learn about this match rule.

1. Open the **Web Adapter Certification** solution.

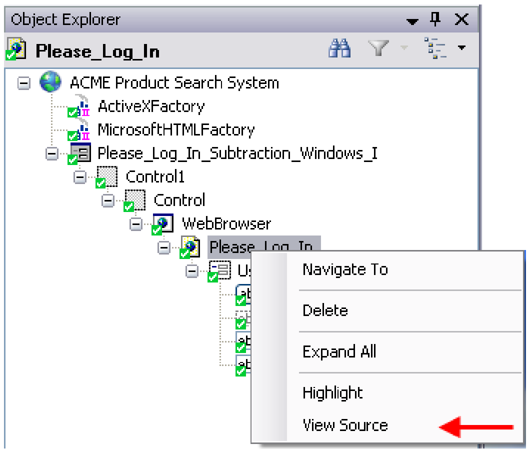
2. Open the **ACME Product Search System** project item in the designer and click the **Start Interrogation** button.

3. In the Object Explorer, select the **user_name** textbox on the **Please_Log_in** page.

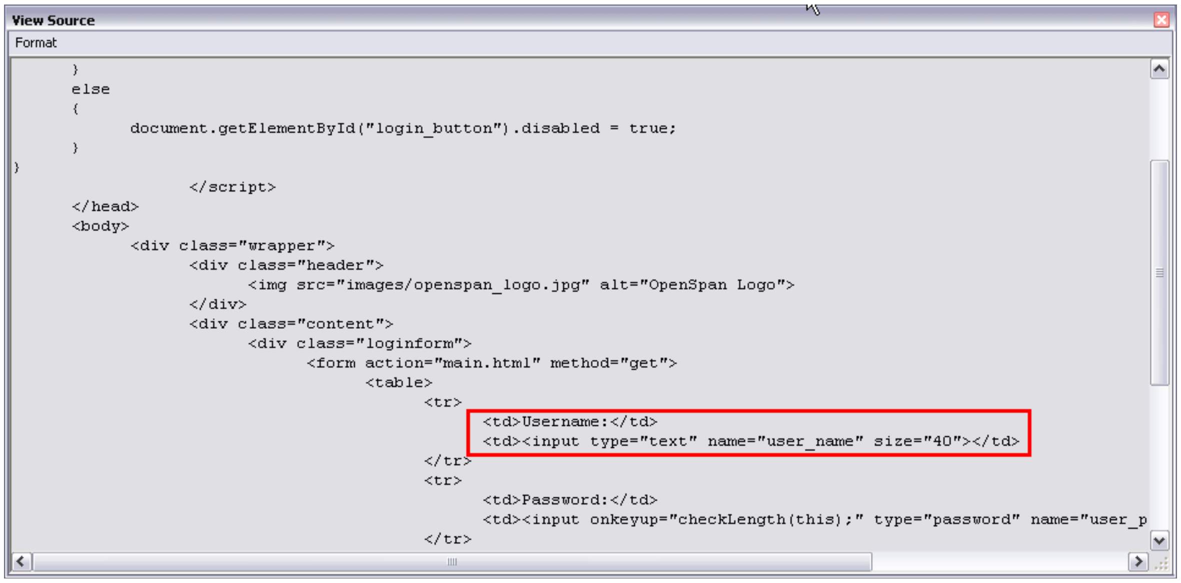
4. Add the **Attribute Value Match Rule** for the **user_name** textbox control by clicking the New button in the Match Rules group box.



5. In the Object Explorer, right-click the **Please_Log_in** web page and select **View Source**.



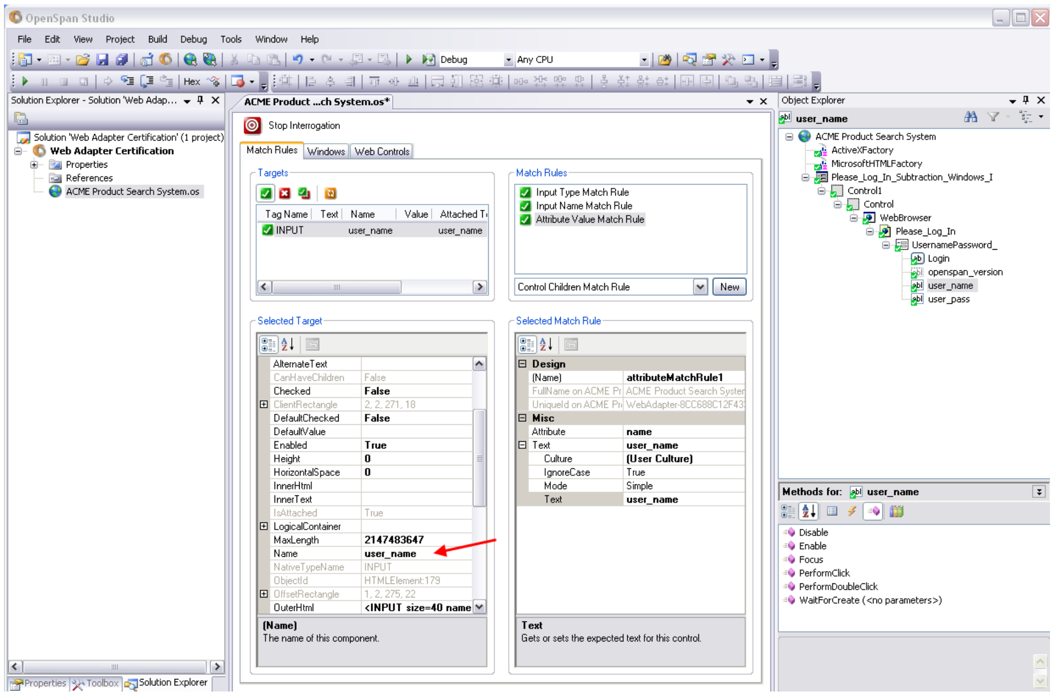
6. Note the attributes surrounding the **user_name** textbox in the following example.



```
        }
        else
        {
                document.getElementById("login_button").disabled = true;
        }
}
                </script>
        </head>
        <body>
                <div class="wrapper">
                        <div class="header">
                                <img src="images/openspan_logo.jpg" alt="OpenSpan Logo">
                        </div>
                        <div class="content">
                                <div class="loginform">
                                        <form action="main.html" method="get">
                                                <table>
                                                        <tr>
                                                                <td>Username:</td>
                                                                <td><input type="text" name="user_name" size="40"></td>
                                                        </tr>
                                                        <tr>
                                                                <td>Password:</td>
                                                                <td><input onkeyup="checkLength(this);" type="password" name="user_p
                                                        </tr>
```

7. Close the **View Source** dialog

8. The same information is available from the **Selected Target** group box while a control is matched.



Tag Name	Text	Name	Value	Attached T
✓ INPUT		user_name		user_name

AlternateText	
CanHaveChildren	False
Checked	False
ClientRectangle	2, 2, 271, 18
DefaultChecked	False
DefaultValue	
Enabled	True
Height	0
HorizontalSpace	0
InnerHtml	
InnerText	
IsAttached	True
LogicalContainer	
MaxLength	2147483647
Name	user_name
NativeTypeName	INPUT
ObjectId	HTMLElement:179
OffsetRectangle	1, 2, 275, 22
OuterHtml	<INPUT size=40 name

(Name)
The name of this component.

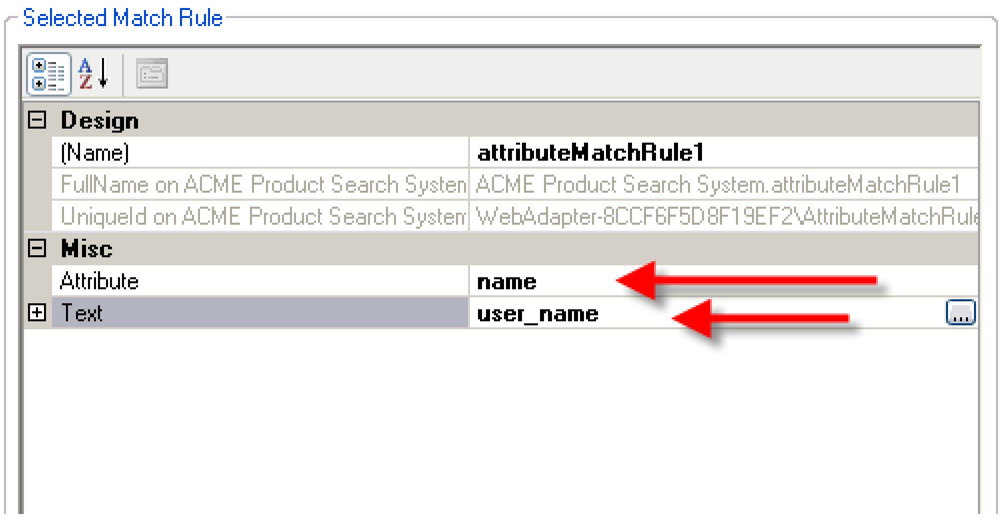
Design	
(Name)	attributeMatchRule1
FullName on ACME Pr	ACME Product Search Syster
UniqueId on ACME Pr	WebAdapter-8CC688C12F43
Misc	
Attribute	name
Text	user_name
Culture	(User Culture)
IgnoreCase	True
Mode	Simple
Text	user_name

Text
Gets or sets the expected text for this control.

9. In the Object Explorer, select the **user_name** textbox.

10. Return to the **Match Rules** tab and select (highlight) the **Attribute Value Match Rule** in the **Match Rules** group box.

11. Enter name in the **Attribute** field and user_name in the **Text** field.

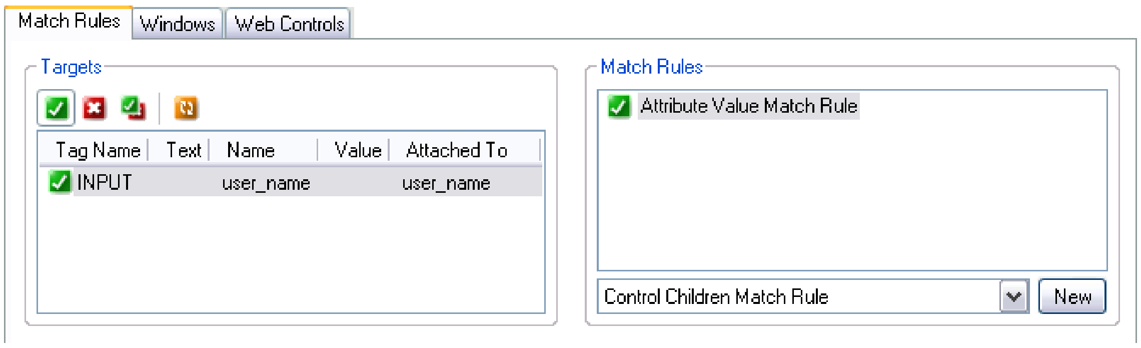


(Name)	attributeMatchRule1
FullName on ACME Product Search Systen	ACME Product Search System.attributeMatchRule1
UniqueId on ACME Product Search Systen	WebAdapter-8CCF6F5D8F19EF2\AttributeMatchRule
Attribute	name
Text	user_name

12. Select the **Refresh Match** button ( ) to determine if the new match rule matches the target as shown in the following image.

13. Delete the default match rules so that only the **Attribute Value Match Rule** remains in the **Match Rules** group box.

14. Confirm that only a single target is matched.



Tag Name	Text	Name	Value	Attached To
✓ INPUT		user_name		user_name

15. Stop the interrogation and save the solution.

Key Default Match Rules

Descriptions of the commonly used default match rules:

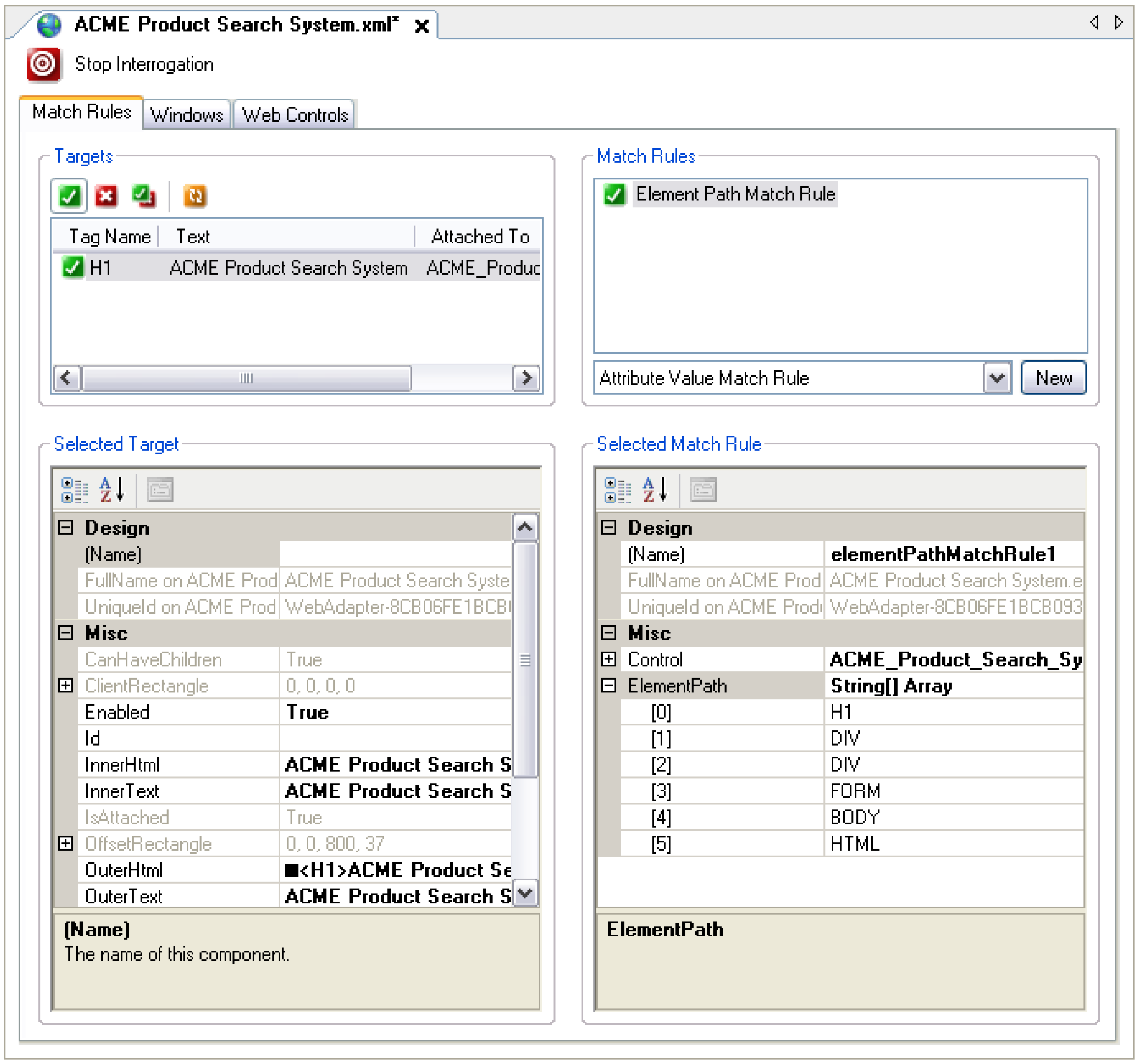
Control Children
When the default match rules fail to uniquely match a target, use the **Control Children Match Rule** to add additional matching constraints. This match rule enables you to define matching of a parent object based on the existence and matching criteria of a child object.

Element ID
Normally, when OpenSpan Studio loads a page and attempts to match the object, OpenSpan Studio queries the browser for a list of web page elements and the tags for the elements appearing on the interrogated page. It then attempts to match based on the tag information. Element ID is the fastest of all match rules and it overrides all other rules applied.

If the element ID match rule fails (because the IDs are dynamic and change every time the page opens or because the Element ID is blank or duplicated) then the object in Object Explorer appears unmatched and there are no other targets listed in the **Targets** group box (since OpenSpan only retrieved Element IDs and not HTML tags).

If this occurs, do not use the **Element ID Match Rule**. You can debug the problem by setting the **Element ID Match Rule, Use Element ID**, property to **False**, doing debug matching and then adding new match rules.

Element Path
The **Element Path Match Rule** is often used when OpenSpan Studio attempts to match a generic web control. For example, when you interrogated the Heading text on the **Login** page of the ACME Product Search System, the **Element Path Match Rule** was applied.



ACME Product Search System.xml*

Stop Interrogation

Match Rules | Windows | Web Controls

Targets

Tag Name	Text	Attached To
✓ H1	ACME Product Search System	ACME_Produc

Match Rules

✓ Element Path Match Rule

Attribute Value Match Rule | New

Selected Target

Design	
(Name)	
FullName on ACME Prod	ACME Product Search Syste
UniqueId on ACME Prod	WebAdapter-8CB06FE1BCB(
Misc	
CanHaveChildren	True
ClientRectangle	0, 0, 0, 0
Enabled	**True**
Id	
InnerHtml	**ACME Product Search S**
InnerText	**ACME Product Search S**
IsAttached	True
OffsetRectangle	0, 0, 800, 37
OuterHtml	**■<H1>ACME Product Se**
OuterText	**ACME Product Search S**

(Name)
The name of this component.

Selected Match Rule

Design	
(Name)	**elementPathMatchRule1**
FullName on ACME Prod	ACME Product Search System.e
UniqueId on ACME Prodi	WebAdapter-8CB06FE1BCB093
Misc	
Control	**ACME_Product_Search_Sy**
ElementPath	**String[] Array**
[0]	H1
[1]	DIV
[2]	DIV
[3]	FORM
[4]	BODY
[5]	HTML

ElementPath

The **ElementPath** property for the **Element Path Match Rule** shows the tags surrounding the selected target. In the case of the heading **ACME Product Search System**, the following tags comprise the **ElementPath**:

- H1
- Div
- Div
- Form
- Body
- HTML

Modifying Match Rules

Document Title and Document URL Match Rules

A common situation in which objects fail to match occurs when the Document Title contains specific information that changes depending on the user interaction with the application. OpenSpan Studio uses the title of the page in the Document Title Match Rule. During interrogation, the page may contain specific information in the title that changes depending on the application use such as:

- User login information contained in title (e.g., user name)
- Parameter value (e.g., Account Number) contained in page title
- System information contained in title (e.g., ACME Software – Test Server)

To remedy any potential match failures, you can modify the matching text used in the Document Title match rule. Click on the more (…) button next to the Text property to open the Match Rule Editor. The editor enables you to change match rule Text through application of the Mode property. The editor includes the following fields:

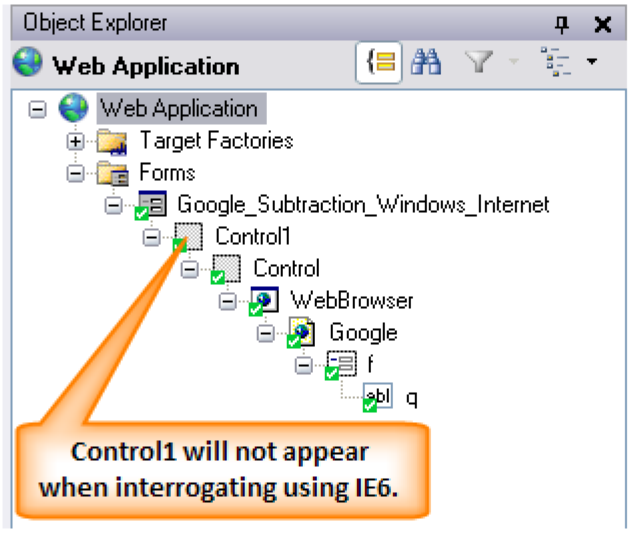
- **Text**: Document title. Modify to support **Mode** selection.
- **Mode**: Default **Simple** – text matched literally. Mode types available include *Regex*, *StartsWith*, *EndsWith*, etc. **Regex** mode indicates that the Text field entry contains Regex syntax.
- **Culture**: Language (culture) used by target application end users. Default **(User Culture)** uses the operating system setting (Control Panel | Regional and Language Options).
- **Ignore Case**: Boolean, defaults checked, which indicates that text can appear in either upper or lower case.
- **Escape**: Applies to Regex mode only. Selecting button prepends Regex escape '\' character to any Regex specific commands contained in the Text field entry.
- **Test**: Enter characters to test match rule. **Not Matched** or **Matched** displays directly above text box as you type.

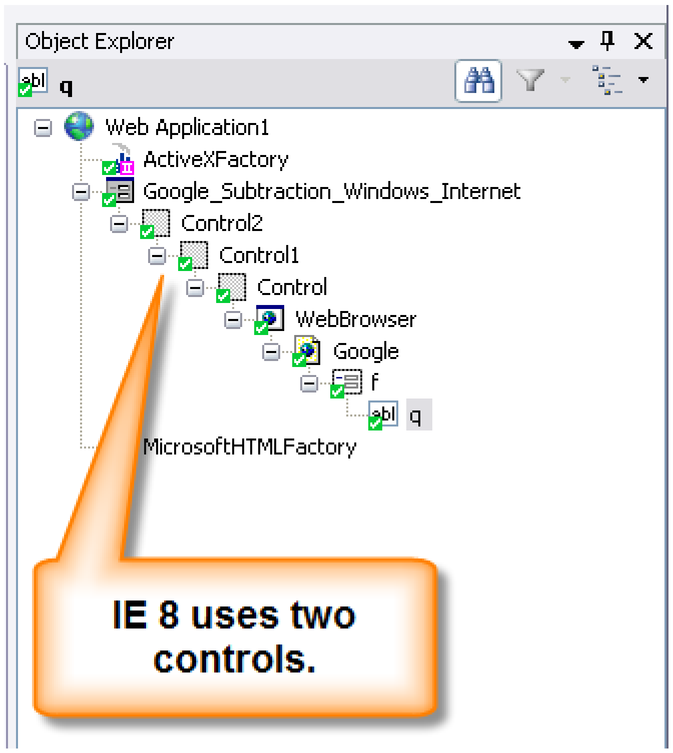
Exercise - Modifying Document Title Match Rule

Use the following steps to modify the Document Title Match Rule:

1. Create a new solution named **Google.**
2. Add a web application project item to the solution.
3. Set the **StartPage** to **www.google.com.**
4. Interrogate the Google search textbox. In the Interrogation Form dialog, use the **Default** option and don't select the **Create Global Web Page** check box.

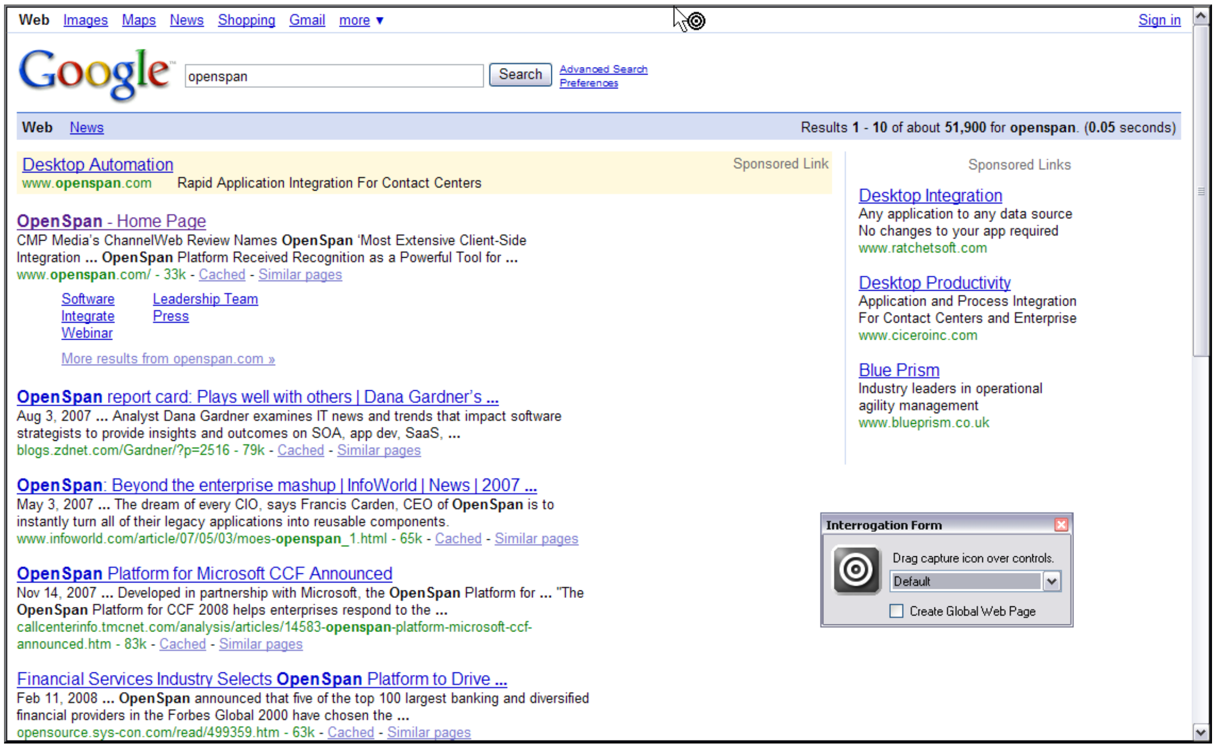
The Object Explorer should appear as follows.





5. Enter a value in the Google search text box, such as *OpenSpan*, and initiate the search. The Google Results page displays.

6. Now interrogate the Google Results page. Select the entire page.



Web Images Maps News Shopping Gmail more ▾ Sign in

Google openspan Search Advanced Search Preferences

Web News Results **1** - **10** of about **51,900** for **openspan**. (**0.05** seconds)

Desktop Automation Sponsored Link
www.**openspan**.com Rapid Application Integration For Contact Centers

OpenSpan - Home Page
CMP Media's ChannelWeb Review Names **OpenSpan** 'Most Extensive Client-Side Integration ... **OpenSpan** Platform Received Recognition as a Powerful Tool for ...
www.**openspan**.com/ - 33k - Cached - Similar pages
Software Leadership Team
Integrate Press
Webinar
More results from openspan.com »

OpenSpan report card: Plays well with others | Dana Gardner's ...
Aug 3, 2007 ... Analyst Dana Gardner examines IT news and trends that impact software strategists to provide insights and outcomes on SOA, app dev, SaaS, ...
blogs.zdnet.com/Gardner/?p=2516 - 79k - Cached - Similar pages

OpenSpan: Beyond the enterprise mashup | InfoWorld | News | 2007 ...
May 3, 2007 ... The dream of every CIO, says Francis Carden, CEO of **OpenSpan** is to instantly turn all of their legacy applications into reusable components.
www.infoworld.com/article/07/05/03/moes-**openspan**_1.html - 65k - Cached - Similar pages

OpenSpan Platform for Microsoft CCF Announced
Nov 14, 2007 ... Developed in partnership with Microsoft, the **OpenSpan** Platform for ... "The **OpenSpan** Platform for CCF 2008 helps enterprises respond to the ...
callcenterinfo.tmcnet.com/analysis/articles/14583-**openspan**-platform-microsoft-ccf-announced.htm - 83k - Cached - Similar pages

Financial Services Industry Selects **OpenSpan** Platform to Drive ...
Feb 11, 2008 ... **OpenSpan** announced that five of the top 100 largest banking and diversified financial providers in the Forbes Global 2000 have chosen the ...
opensource.sys-con.com/read/499359.htm - 63k - Cached - Similar pages

Sponsored Links

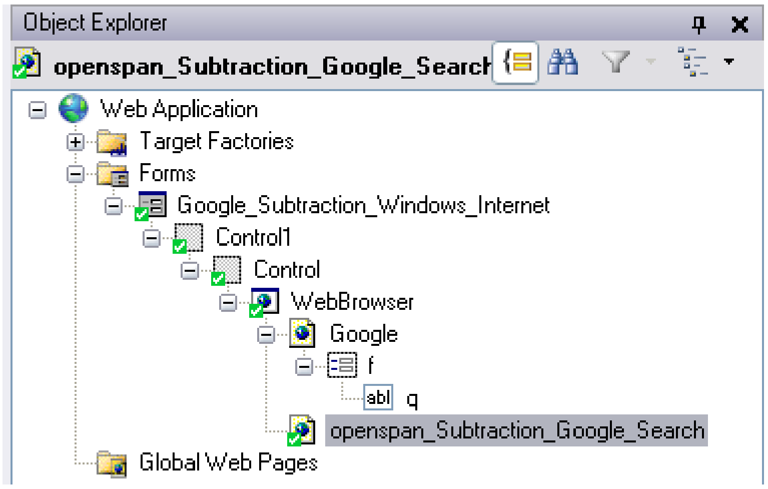
Desktop Integration
Any application to any data source
No changes to your app required
www.ratchetsoft.com

Desktop Productivity
Application and Process Integration
For Contact Centers and Enterprise
www.ciceroinc.com

Blue Prism
Industry leaders in operational agility management
www.blueprism.co.uk

Interrogation Form
Drag capture icon over controls.
Default
☐ Create Global Web Page

Your Object Explorer should look like this:



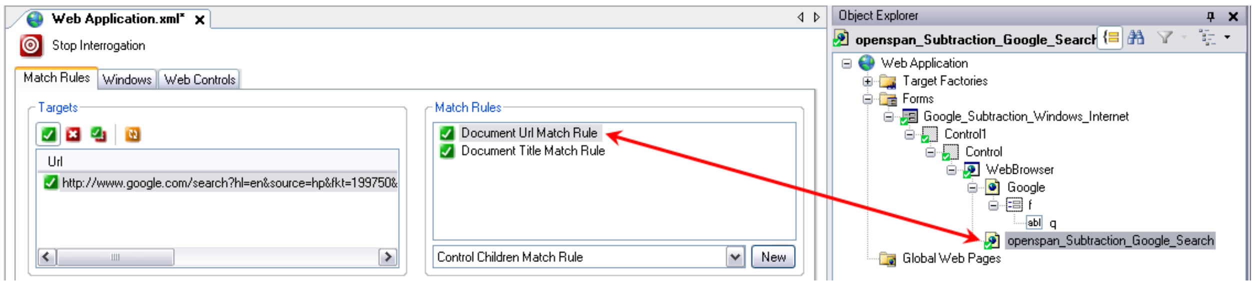
Object Explorer
openspan_Subtraction_Google_Search

```
Web Application
    Target Factories
    Forms
        Google_Subtraction_Windows_Internet
            Control1
                Control
                    WebBrowser
                        Google
                            f
                                q
                        openspan_Subtraction_Google_Search
    Global Web Pages
```

Note: IE 8 will display Controls1 and 2.

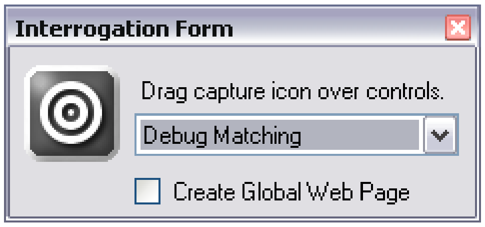
Note that the match rules for the result page (**Openspan_Subtraction_Google_Search**) are:

- **Document Url Match Rule**: http://www.google.com/search?hl=en&q=openspan&btnG=Google+Search
- **Document Title Match Rule**: openspan – Google Search



7. Navigate back to the Google home page – www.google.com. Initiate a new search using another term, for example: *frogs*. Note that the page and child objects of the page no longer match in the Object Explorer.

8. To determine the cause of the match failure, select the **Debug Matching** option in the Interrogation Form.


Interrogation Form

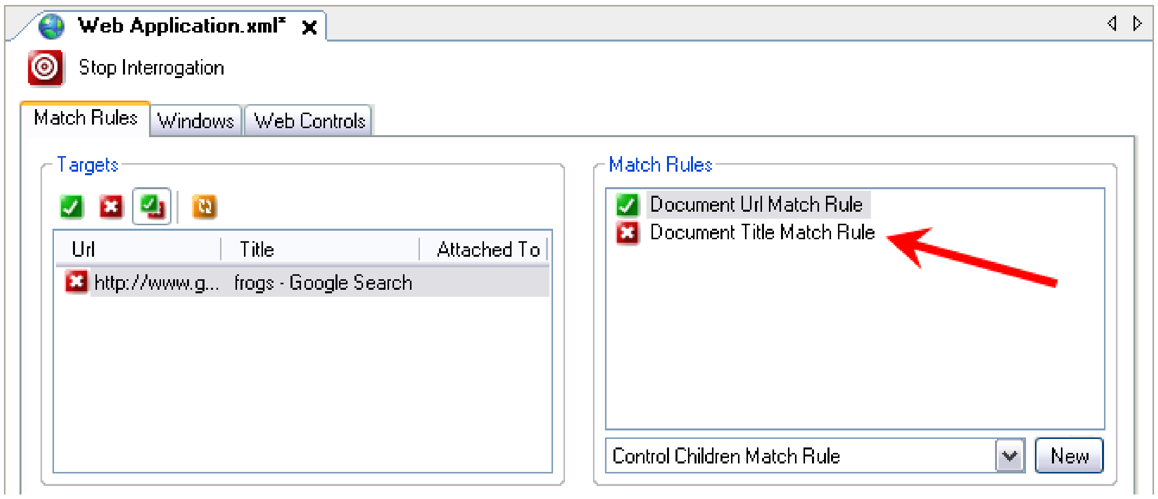
Drag capture icon over controls.

Debug Matching

☐ Create Global Web Page

9. With the **Openspan_Subraction_Google_Search** web page selected in the **Object Explorer**, drag the target icon over the Google Results page.

10. In the **Match Rules** pane, the status of the rules display as shown in the following image:



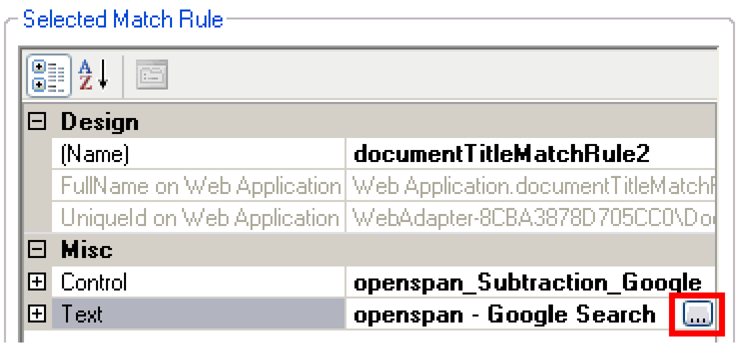
Note that the match rules for the result page are:

- Document URL: http://www.google.com/search?hl=en&q=frogs&btnG=Google+Search
- Document Title: openspan – Google Search

While the Query part of the URL differs, the Document URL Match Rule remains matched. By default, OpenSpan only uses the information up to the Query portion of the URL to determine the match: http://www.google.com/search.

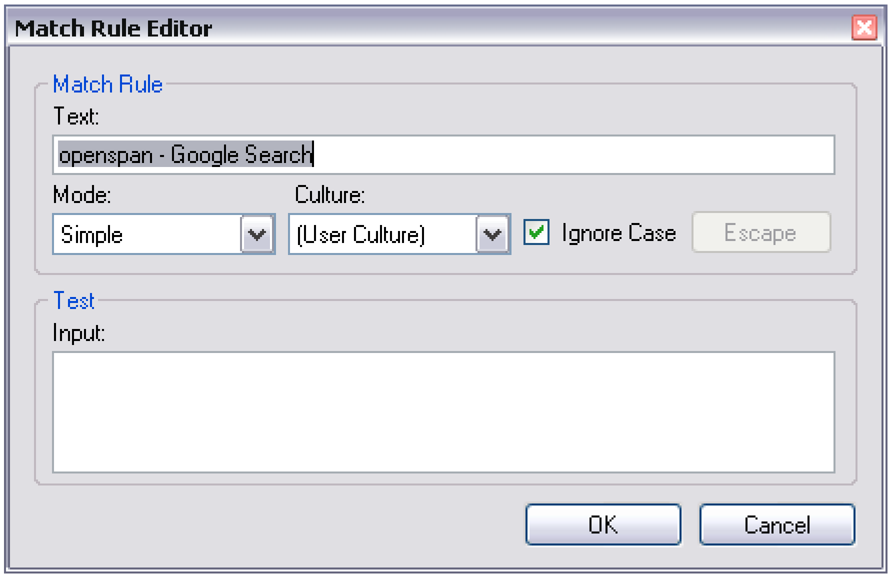
However, the Document Title Match Rule uses the full text of the title *frogs – Google Search*. Since the initial page title was *openspan – Google Search* it no longer matches the current title *frogs – Google Search*.

11. To remedy the matching failure, highlight the **Document Title Match Rule** in the **Match Rules** group box. Click on the browse button in the Text property in the **Selected Match Rule** group box.



Design	
(Name)	**documentTitleMatchRule2**
FullName on Web Application	Web Application.documentTitleMatchR
UniqueId on Web Application	WebAdapter-8CBA3878D705CC0\Do
Misc	
Control	**openspan_Subtraction_Google**
Text	**openspan - Google Search**

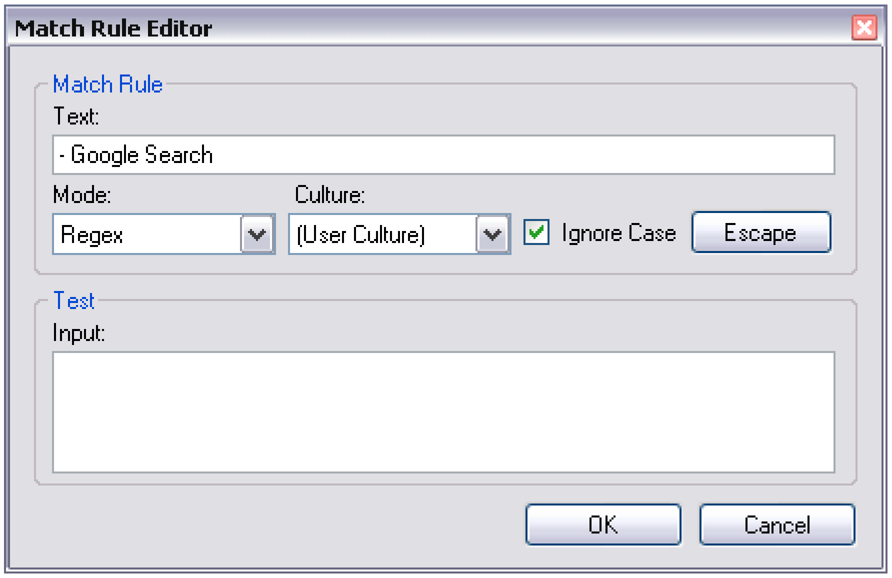
12. The Match Rule Editor displays. An example follows:


Match Rule Editor

Match Rule
Text: openspan - Google Search
Mode: Simple
Culture: (User Culture)
☑ Ignore Case

Change the Mode to **Regex**.

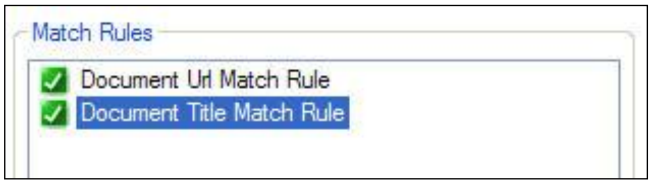
13. In the **Text** textbox, delete the leading text (e.g., openspan) leaving - *Google Search*. This will cause the system to match whenever it sees the text - *Google Search*. Your selected match rule should look like the following image:


Match Rule Editor

Match Rule
Text: - Google Search
Mode: Regex
Culture: (User Culture)
☑ Ignore Case

14. Click OK to save the changes.

15. Refresh the matching status. Now both the match rules appear matched.



✓ Document Url Match Rule
✓ Document Title Match Rule

16. Navigate back to the Google home page one more time – www.google.com. Initiate a new search using another term, for example: *Artie's Deli*. Note that the page and child objects of the page continue to match in the Object Explorer.

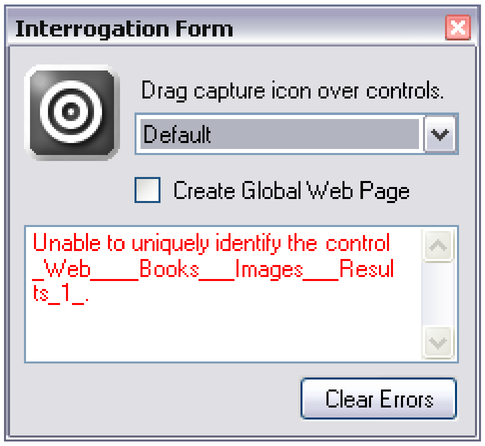
17. Save and close the solution.

Ambiguous Matching

Multiple matches occur when the default match rules fail to *uniquely* identify a single target. This usually occurs because the information available about the target is insufficient to differentiate it from other targets in the application.

Controls

OpenSpan attempts to identify targets precisely. For example, when you interrogate a submit button in a web application, OpenSpan attempts to identify the object as an **INPUT** object with tags for **text** and **type**. In some cases, the targets can only be defined in OpenSpan as generic web controls. When interrogating targets that OpenSpan identifies as generic Web Controls, multiple matching targets may be identified. If this occurs, you are notified of the issue by a message appearing the in the Interrogation Form.



Unable to uniquely identify the control _Web____Books___Images___Results_1_.

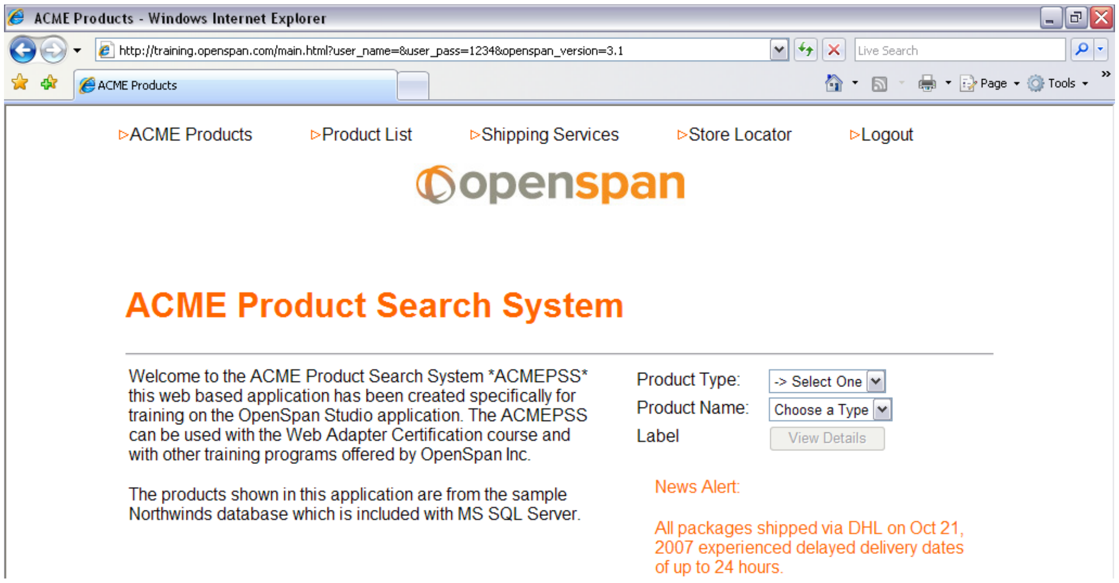
To remedy multiple target matches, you can add match rules to further restrict matching parameters, or you can modify the default match rule to more precisely match the desired target. The following exercise demonstrates how to add new match rules to remedy multiple target matches:

Exercise - Modifying Element Inner Text Match Rule

1. Open the **Web Adapter Certification** solution.

2. Open the **ACME Product Search System** project item in the designer and click the **Start Interrogation** button.

3. In the **Login** screen, enter 1234 in the **Password** field (Note: **Username** field entry is not required) and select the **Login** button. The **Acme Product Search System** screen opens.


ACME Products - Windows Internet Explorer
http://training.openspan.com/main.html?user_name=&user_pass=1234&openspan_version=3.1
Live Search
ACME Products
Page
Tools
▹ACME Products
▹Product List
▹Shipping Services
▹Store Locator
▹Logout
openspan

ACME Product Search System

Welcome to the ACME Product Search System *ACMEPSS* this web based application has been created specifically for training on the OpenSpan Studio application. The ACMEPSS can be used with the Web Adapter Certification course and with other training programs offered by OpenSpan Inc.

The products shown in this application are from the sample Northwinds database which is included with MS SQL Server.

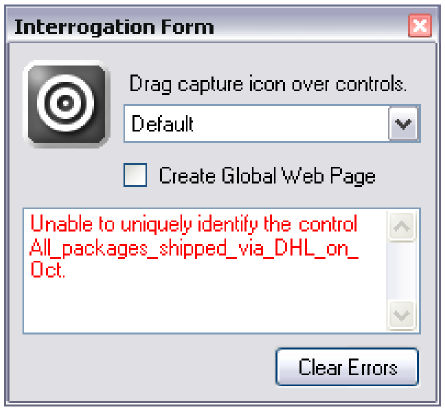
Product Type: -> Select One
Product Name: Choose a Type
Label
View Details

News Alert:

All packages shipped via DHL on Oct 21, 2007 experienced delayed delivery dates of up to 24 hours.

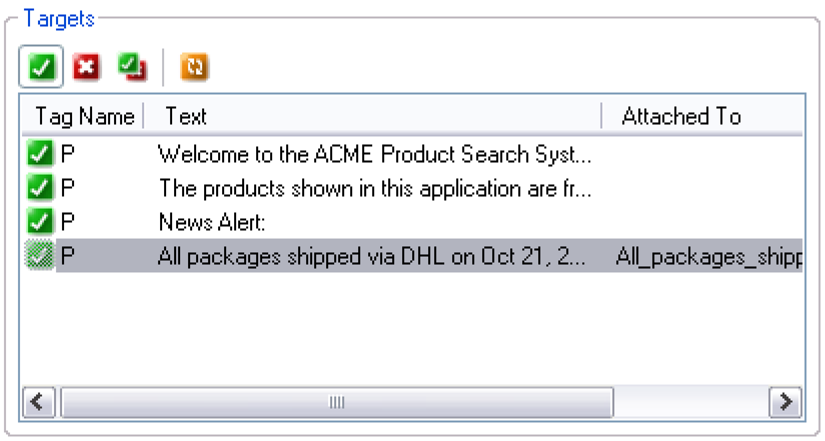
4. Next, interrogate News Alert text: **All packages shipped via DHL on Oct 21, 2007 experienced delayed delivery dates of up to 24 hours.**

5. The **Interrogation Form** will display the message **Unable to uniquely identify the control...**



Unable to uniquely identify the control All_packages_shipped_via_DHL_on_Oct.

6. Inspect the **Targets** group box on the **Match Rules** tab and note that multiple Paragraph type elements are matched:



Tag Name	Text	Attached To
✓ P	Welcome to the ACME Product Search Syst...	
✓ P	The products shown in this application are fr...	
✓ P	News Alert:	
✓ P	All packages shipped via DHL on Oct 21, 2...	All_packages_shipp

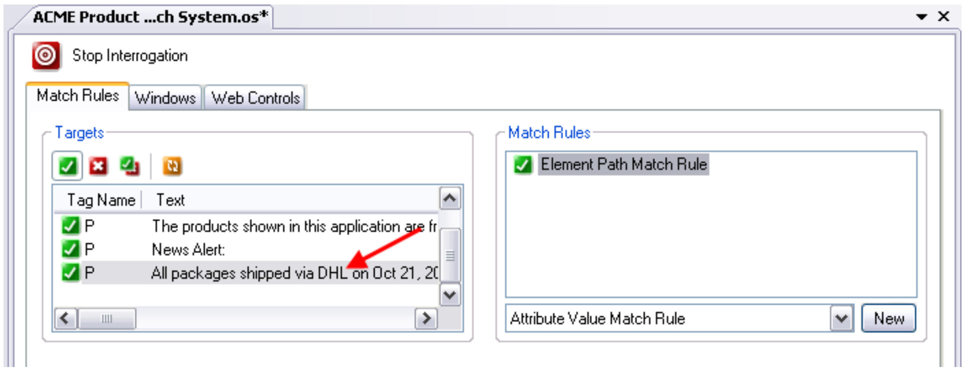
The default match rule applied for this target is the **Element Path Match Rule**. The tags for the headings are identical:

- P
- TD
- TR
- TBODY
- TABLE
- DIV
- DIV
- BODY
- HTML

In this case, the best way to remedy the multiple matches is to add a new match rule that contains parameters to distinguish the two heading elements. The available options are:

- Attribute Value
- Control Children
- Element ID
- Element Index
- Element Inner Text

7. Highlight the **All packages shipped via DHL .....** target in the **Targets** group box on the **Match Rules** tab.

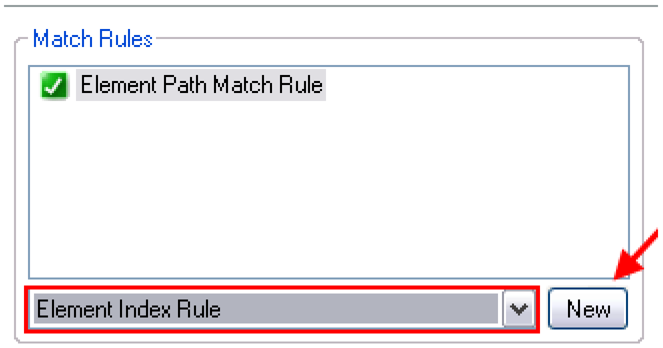


Tag Name	Text
P	The products shown in this application are fr
P	News Alert:
P	All packages shipped via DHL on Oct 21, 20

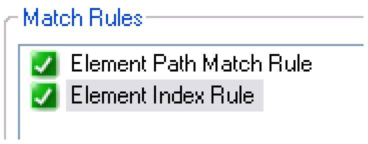
Element Path Match Rule

Attribute Value Match Rule

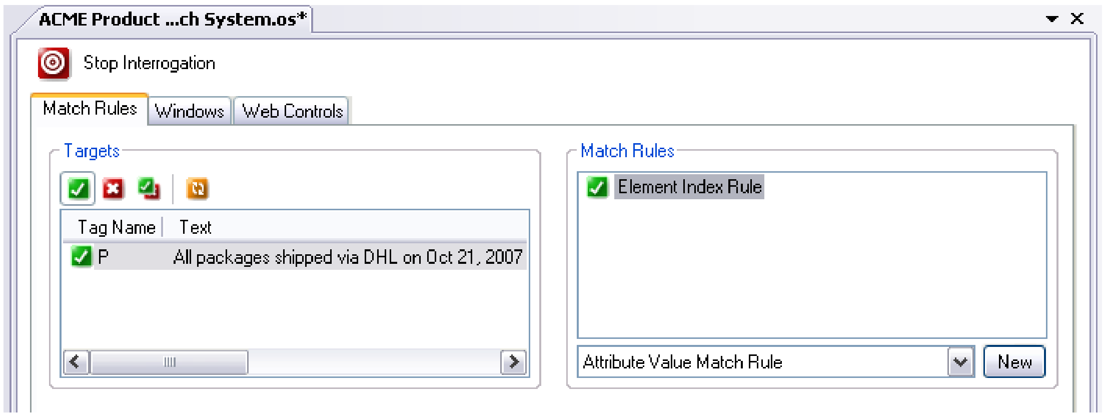
8. Select the **Element Index Rule** from the match rules list and then click the **New** button.



9. The match rule is added to the **Match Rules** group box.



10. Delete the **Element Path** match rule. Note that the heading is now uniquely matched.



Tag Name	Text
P	All packages shipped via DHL on Oct 21, 2007

Match Rules: Element Index Rule

Attribute Value Match Rule

Stop the Interrogator and save the solution.

This page intentionally left blank.

CHAPTER 4: WEB APPLICATION AUTOMATIONS

A typical solution involving web applications requires automating the navigation of the application to a particular page and then performing actions on the target page. This chapter describes methods for navigating to specific pages within a site and confirming that the page is matched prior to performing actions on the page. Topics include:

- Navigating Web Browser to a URL
- Checking Web application state and page
- Using the **IsCreated** Property and the **WaitForCreate** method
- Using Page Links

Automating Login

A common task performed by OpenSpan Studio solutions is to automate login. A user must often enter login information into multiple applications. One way to streamline the workflow is to collect login information in a form and then use the information to login to multiple applications. Important points:

- Verify that login page is open and fully loaded in the browser
- Setting data links to *Sensitive* for passing login information
- Re-starting the application if closed
- Verifying objects are created

In the following exercise, you will get a chance to practice the points listed above.

Exercise – Login Form

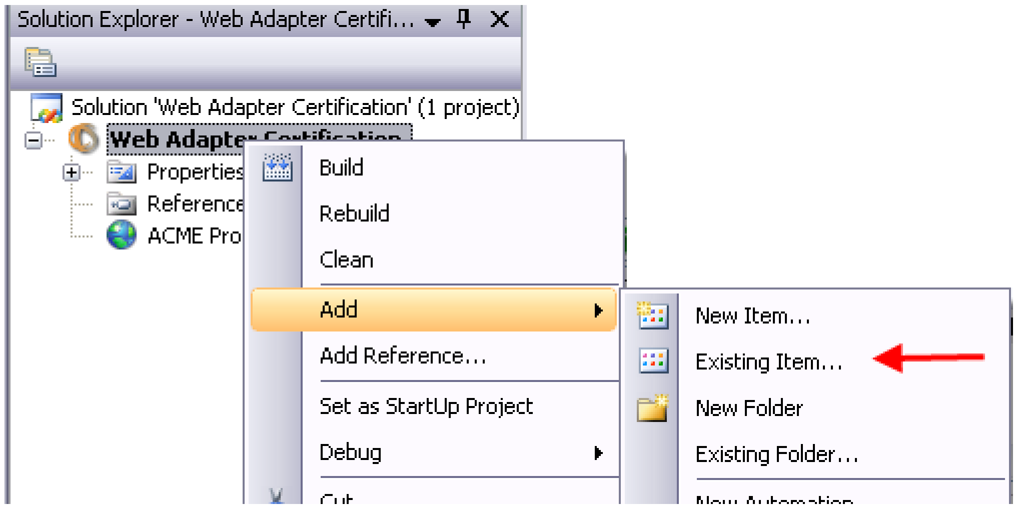
In this exercise you will add a Windows form to capture login information and then send the login credentials to the OpenSpan Training web site. In reality, you do not need a separate form for logging into a single site. However, OpenSpan can be used to create a single form for collecting login credentials for multiple applications within a solution. For the purpose of this course, a single application is used. The design considerations highlighted by this exercise are:

- Verifying that the web site is open prior to sending login information. Since users can close the browser, it is important to check that the browser is running and that the page is set to the web training site Login page.
- Verifying that all controls on the website Login page are fully loaded before sending login information to the page. If the page is currently loading, and therefore the objects are not matched, sending data and events to the page will fail.
- Setting appropriate links to Sensitive so that login information is not recorded in the logs.

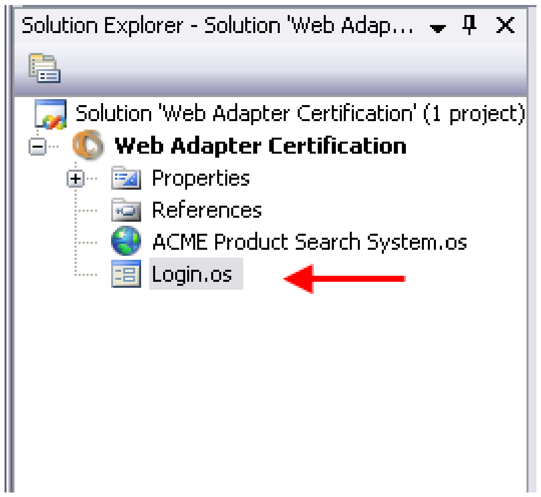
Create Login Window

1. Open the **Web Adapter Certification** solution you created in chapter 2 of this training module.

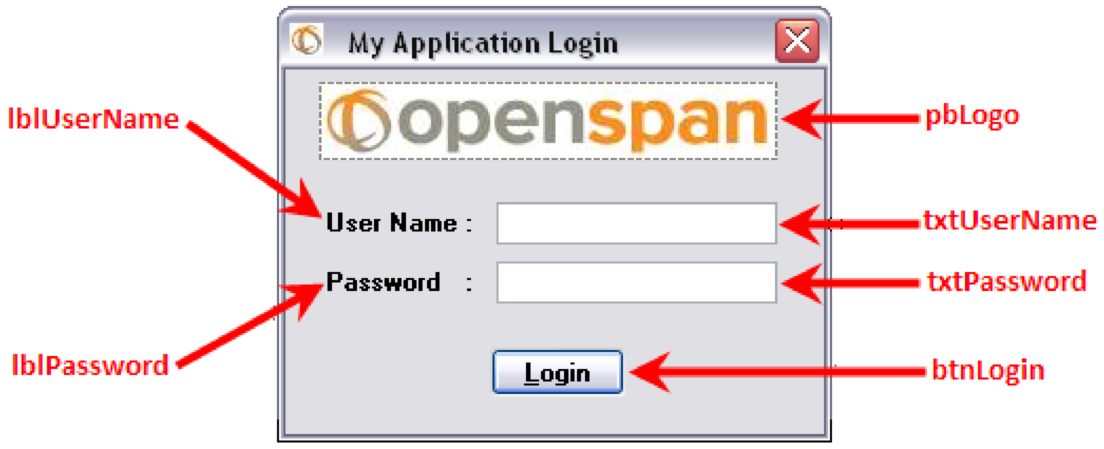
2. In the **Solution Explorer**, right-click the Project and select **Add | Existing Item**.



3. Navigate to the **Extras\Training Solutions** sub-folder of the **OpenSpan Studio** program folder and select the **Login.os** file. The new project item **Login** appears in the **Solution Explorer**:


Solution Explorer - Solution 'Web Adap...
Solution 'Web Adapter Certification' (1 project)
Web Adapter Certification
Properties
References
ACME Product Search System.os
Login.os

4. Open the **Login** form project item. The form contains textboxes for entry of **User Name** and **Password**.



5. Select **File | Save All**.

Add Login Automations

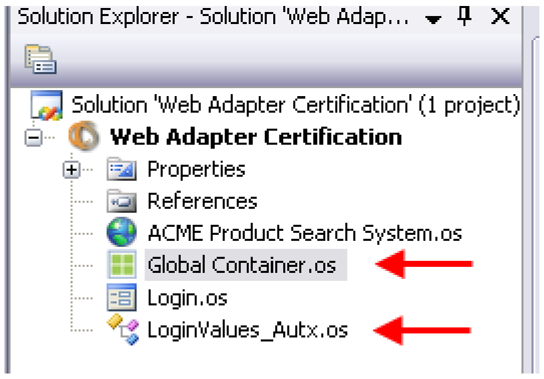
Once you have created a form for users to supply login information, the next steps are to establish the logic for passing login information from the form to the web application. The automation logic needs to accomplish the following:

- Verify that the **User Name** and **Password** textboxes have entries (not null).
- Pass the login information to the **Login** page of the ACME Product Search System web application.
- Ensure that the ACME Product Search System web application is running. Start the application if it is not running.
- Ensure that the correct page is loaded in the browser. Navigate to the **Login** page.
- Ensure that the target textboxes on the **Login** page have been loaded (created) in the page.

1. To validate the User Name and Password information entered on the **Login** form (**My Application Login** window), add the following two items to the solution:

   - **Automation - LoginValues_Autx** – Verifies **User Name** and **Password** field entries are not null and that the password is at least 4 characters in length.
   - **Global Container -** Holds components that are shared throughout the solution. By adding shared components to a Global Container, you can avoid the error of deleting a component unintentionally.

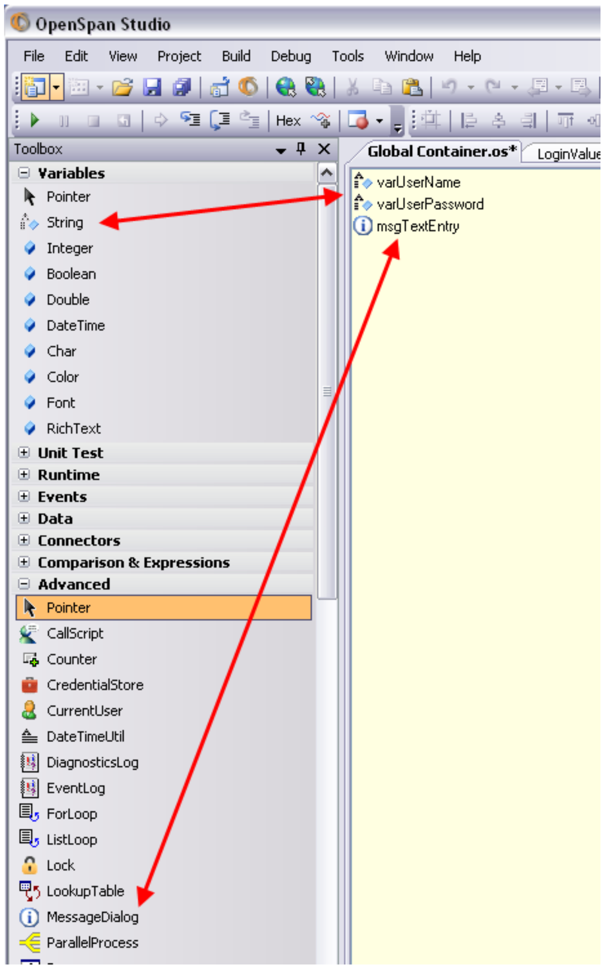
The **Solution Explorer** should look like this:


Solution Explorer - Solution 'Web Adap...
Solution 'Web Adapter Certification' (1 project)
Web Adapter Certification
Properties
References
ACME Product Search System.os
Global Container.os
Login.os
LoginValues_Autx.os

2. Open the **Global Container** item.

3. Add and name the following components:

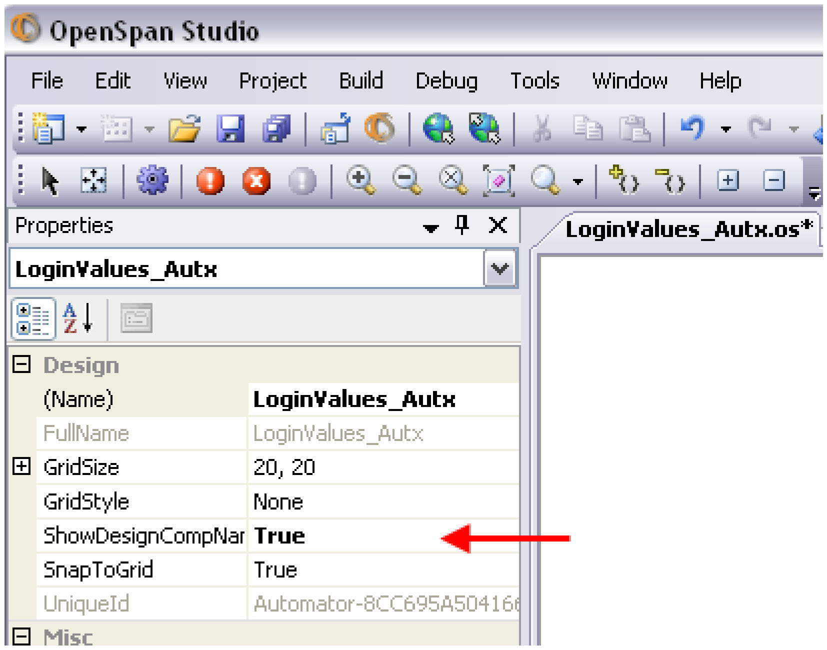
   - (2) String (from Toolbox **Variables** tab): **varUsername** and **varUserPassword**
   - (1) MessageDialog (String message):Dialog Result (from the Toolbox **Advanced** tab): **msgTextEntry**


OpenSpan Studio
File Edit View Project Build Debug Tools Window Help
Hex
Toolbox
Variables
Pointer
String
Integer
Boolean
Double
DateTime
Char
Color
Font
RichText
Unit Test
Runtime
Events
Data
Connectors
Comparison & Expressions
Advanced
Pointer
CallScript
Counter
CredentialStore
CurrentUser
DateTimeUtil
DiagnosticsLog
EventLog
ForLoop
ListLoop
Lock
LookupTable
MessageDialog
ParallelProcess
Global Container.os*
LoginValue
varUserName
varUserPassword
msgTextEntry

4. Select **File | Save All**.

5. Open the **LoginValues_Autx** automation.

6. In the **Properties** window, select the **LoginValues_Autx** automation and change the **ShowDesignCompNames** property to **True**.



LoginValues_Autx

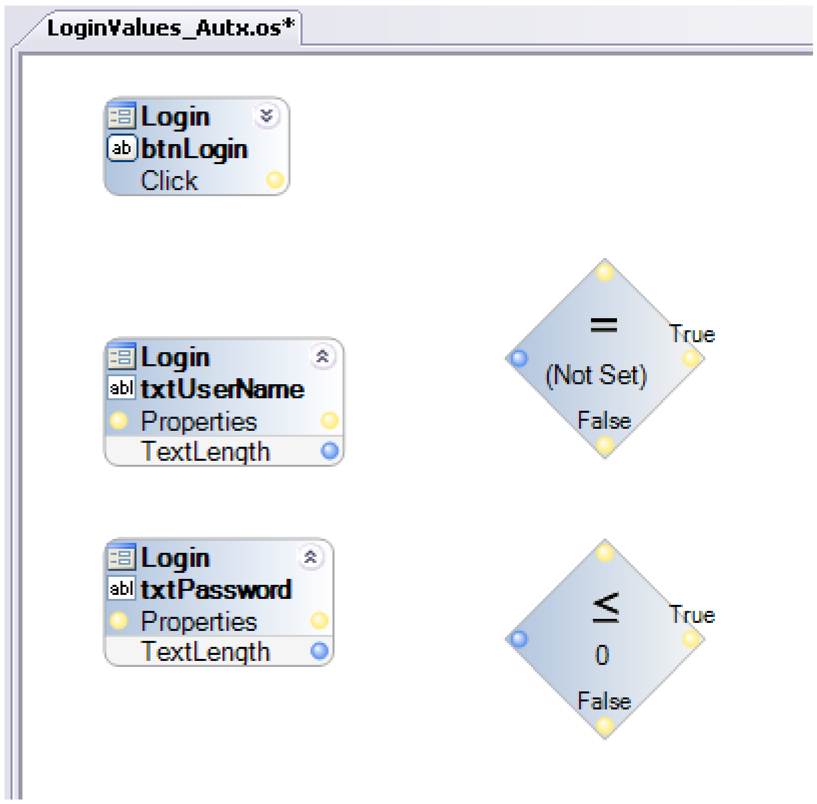
Design	
(Name)	LoginValues_Autx
FullName	LoginValues_Autx
GridSize	20, 20
GridStyle	None
ShowDesignCompNar	True
SnapToGrid	True
UniqueId	Automator-8CC695A50416
Misc	

7. Add the following components to the **LoginValues_Autx** automation:

Component	Source Project Item	Property	Event	Method
btnLogin	Login		Click	
txtUserName	Login	TextLength		
txtPassword	Login	TextLength		
Equals comparison block	Toolbox/Comparison & Expressions tab			
Less than or Equal to comparison block	Toolbox/Comparison & Expressions tab			

If TextLength is not listed at the bottom of the **Object Explorer**, select the **Configure Type** button to add it.

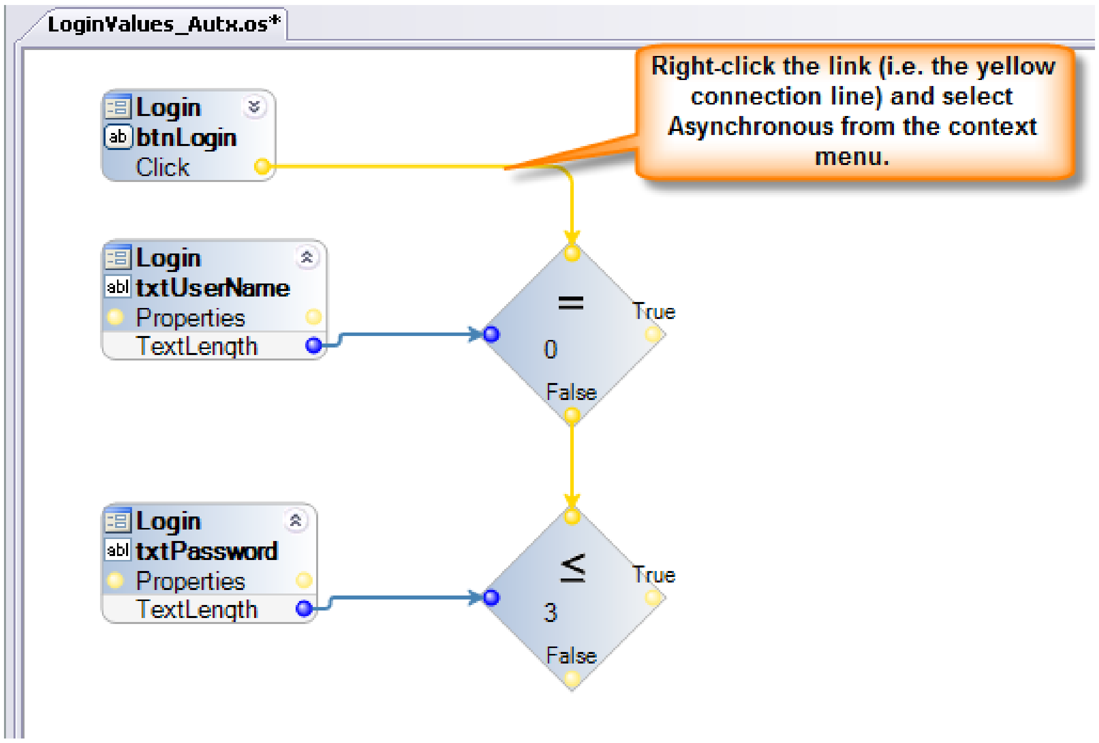
Your automation should appear as follows:



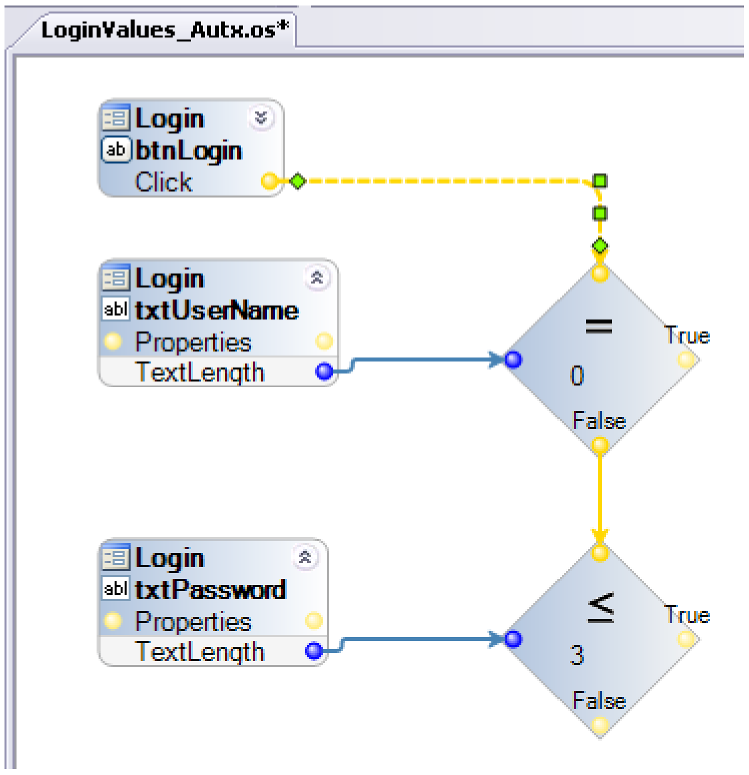
8. Connect the components as follows:

Source Component	Connect To	Target Component
btnLogin.Click execution output	→	Equals comparison block.execution input
txtUserName.TextLength data output	→	Equals comparison block.data input Set comparison block value to 0.
Equals comparison block.False execution output	→	Less Than or Equal to comparison block.execution input
txtPassword.TextLength data output	→	Less Than or Equal to comparison block.data input Set comparison block value to 3 (since the password must contain at least 4 characters for the Login button to become enabled and the Text.Length property initiates counting at 0).

9. Set the event link from the **btnLogin.Click** event to asynchronous by right-clicking the link between the **btnLogin** button and the **Equal to** comparison block and selecting **Asynchronous** from the context menu. By doing this the subsequent solution logic can proceed without locking the Windows form.



Your automation should appear as follows:



10. Add the following additional components to the **LoginValues_Autx** automation:

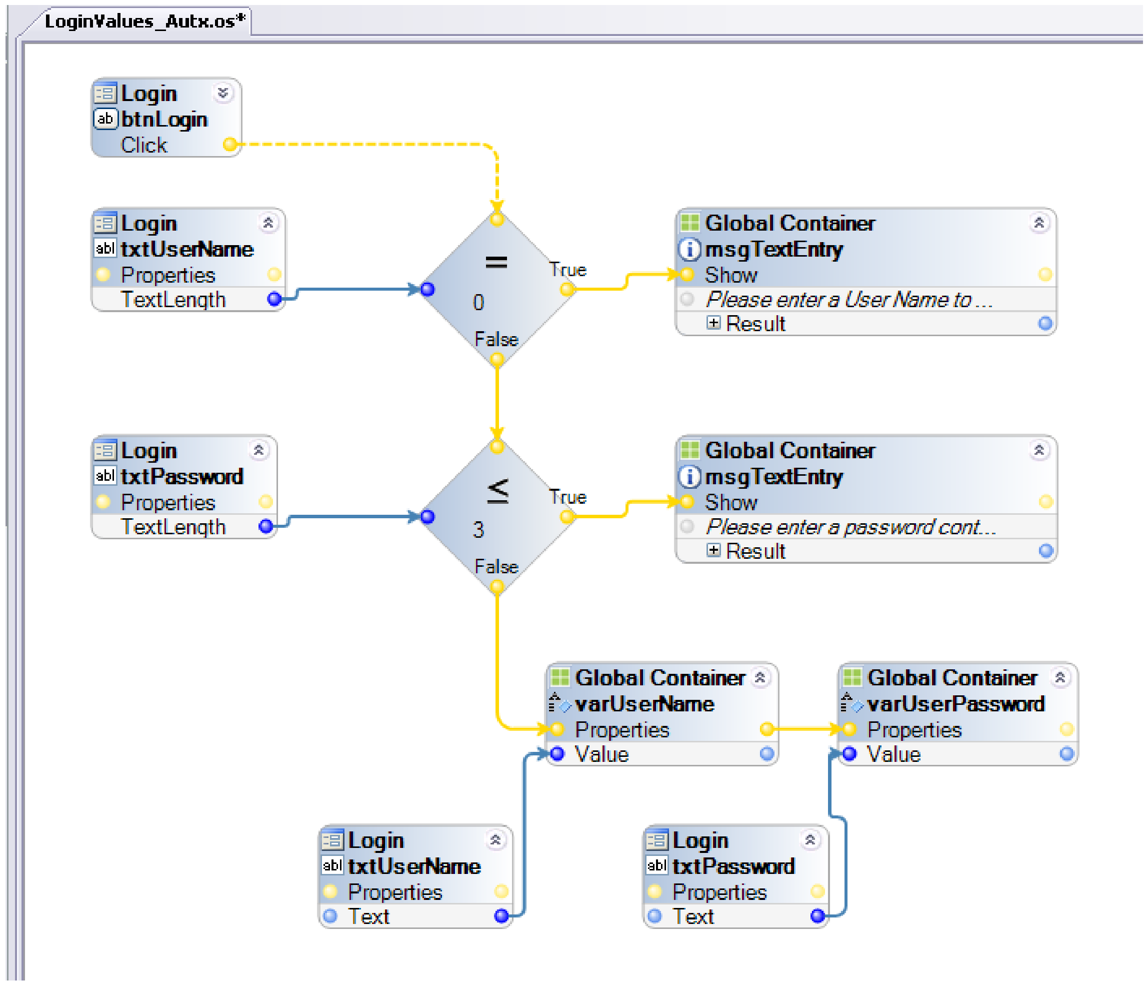
Component	Source Project Item	Property	Event	Method
txtUserName	Login	Text		
txtPassword	Login	Text		
msgTextEntry.Show (1) parameter. Enter message text: **Please enter a User Name to continue.**	Global Container			
msgTextEntry.Show (1) parameter. Enter message text: **Please enter a password containing at least four characters.**	Global Container			
varUserName	Global Container	Value		
varUserPassword	Global Container	Value		

11. Connect the components as follows:

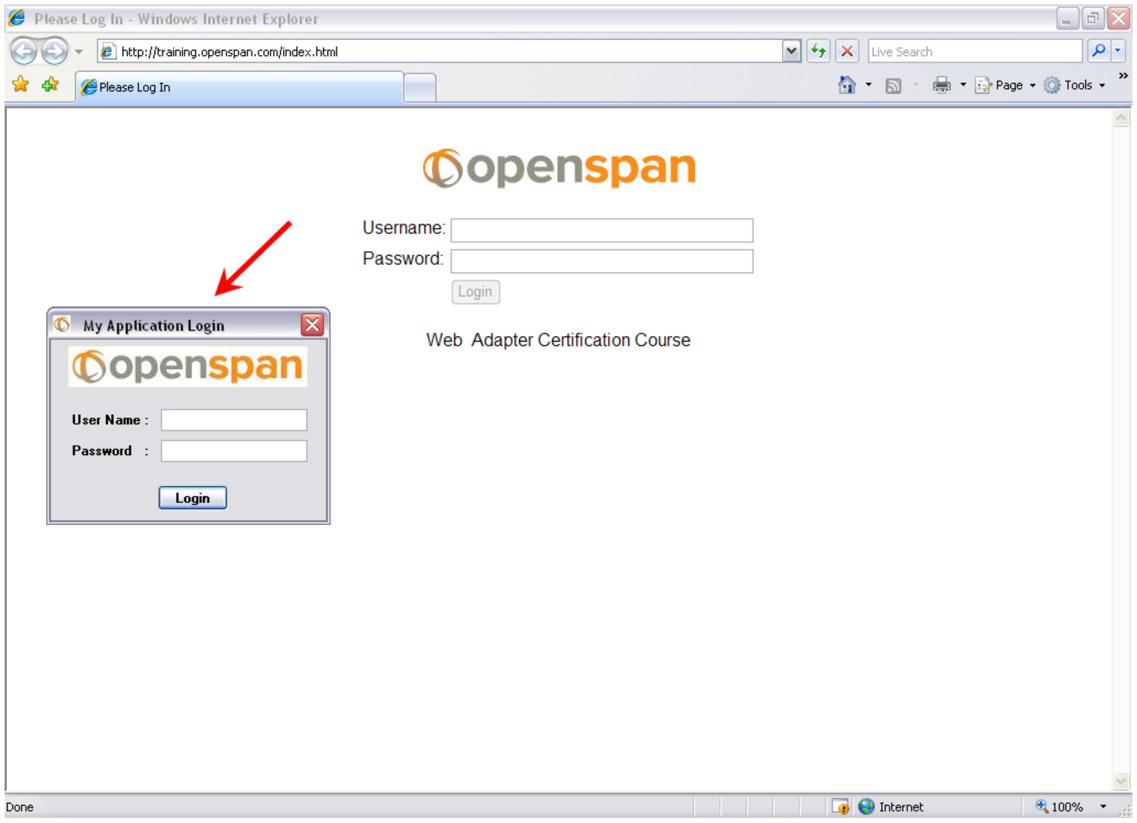
Source Component	Connect To	Target Component
Equals comparison block.True execution output	→	msgTextEntry.execution input. Message text: **Please enter a User Name to continue.**

Less Than or Equal to comparison block.True execution output	→	msgTextEntry.execution input. Message text: **Please enter a password containing at least four characters.**
Less Than or Equal to comparison block.False execution output	→	varUserName.Properties execution Input
txtUserName.Text data output	→	varUserName.Value data Input
varUserName.Properties execution output	→	varUserPassword.Properties execution Input
txtPassword.Text data output	→	varUserPassword.Value data Input

Your automation should look like the following:



12. Save and run the solution. The **My Application Login** window and the ACME Product Search System **Login** page should open:



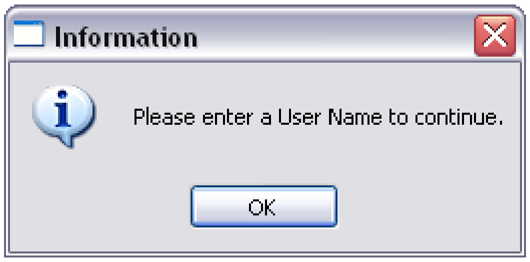
13. Leave the solution running to complete the following test scenarios.

Test Login Automation - Part 1

Test your automation logic with the following scenarios:

- Leave the **User Name** and **Password** fields blank on the **My Application Login** window and click the **Login** button.

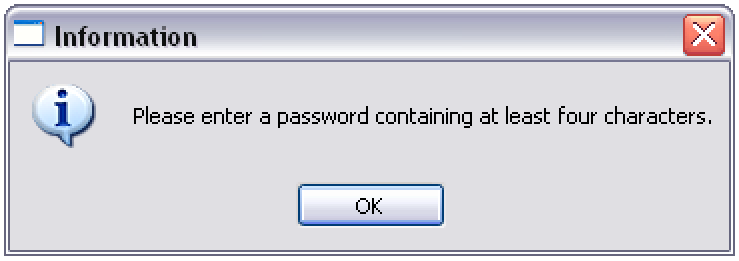
  You should receive a message prompting you to enter the User Name:

  Please enter a User Name to continue.

- Enter a User Name and leave the Password blank on the **My Application Login** window and click the **Login** button.

  You should receive a message prompting you to enter the Password:

  Please enter a password containing at least four characters.

Once you have completed testing this part of the solution, stop the solution and return to OpenSpan Studio.

Add Web Application Verification Logic

The first part of the **LoginValues_Autx** automation verifies that the user enters values in the **User Name** and **Password** fields. Before passing the login information to the web application, you need to enter logic to ensure that the web application is running and the correct page is loaded in the browser.

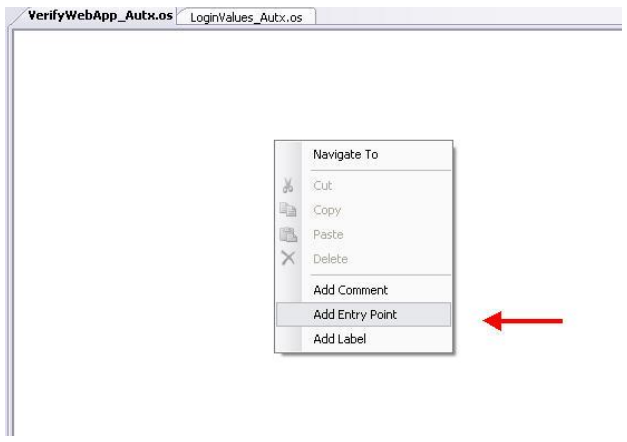
1. Add a new automation to the solution **VerifyWebApp_Autx**.

2. Set the **ShowDesignCompNames** property to **True** for the **VerifyWebApp_Autx** automation.

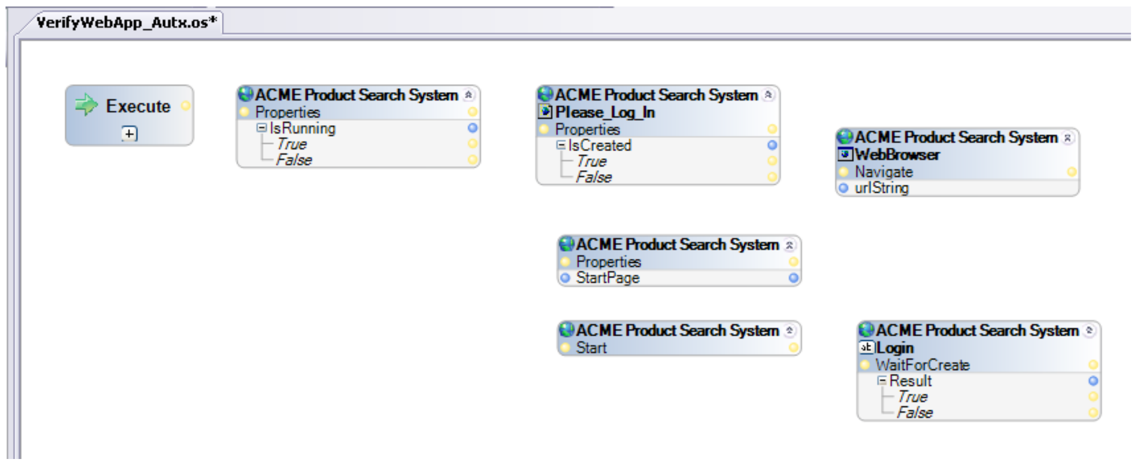
3. Add the following components to the **VerifyWebApp_Autx** automation.

Component	Source Project Item	Property	Event	Method
Execute (entry point)	Right-click in design pane and select Add Entry Point from the context menu to add the Execute entry point.			
Please_ Log_In web page	ACME Product Search System	IsCreated		
ACME Product Search System web application	ACME Product Search System. This is top item in the Object Explorer hierarchy.	IsRunning; StartPage		Start
Web Browser component	ACME Product Search System			Navigate (1 parameter)
Login button	ACME Product Search System.Please_ Log_In web page.			WaitForCreate(<no parameters>)

Entry Point:



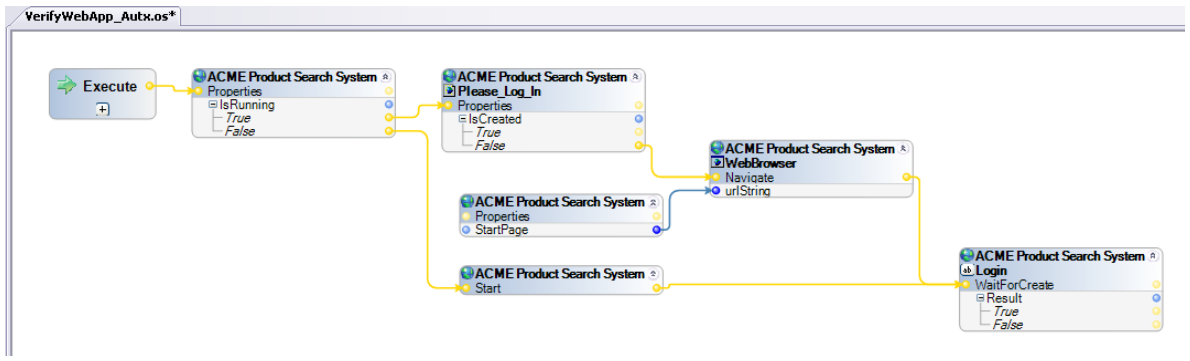
Your automation should contain following components:



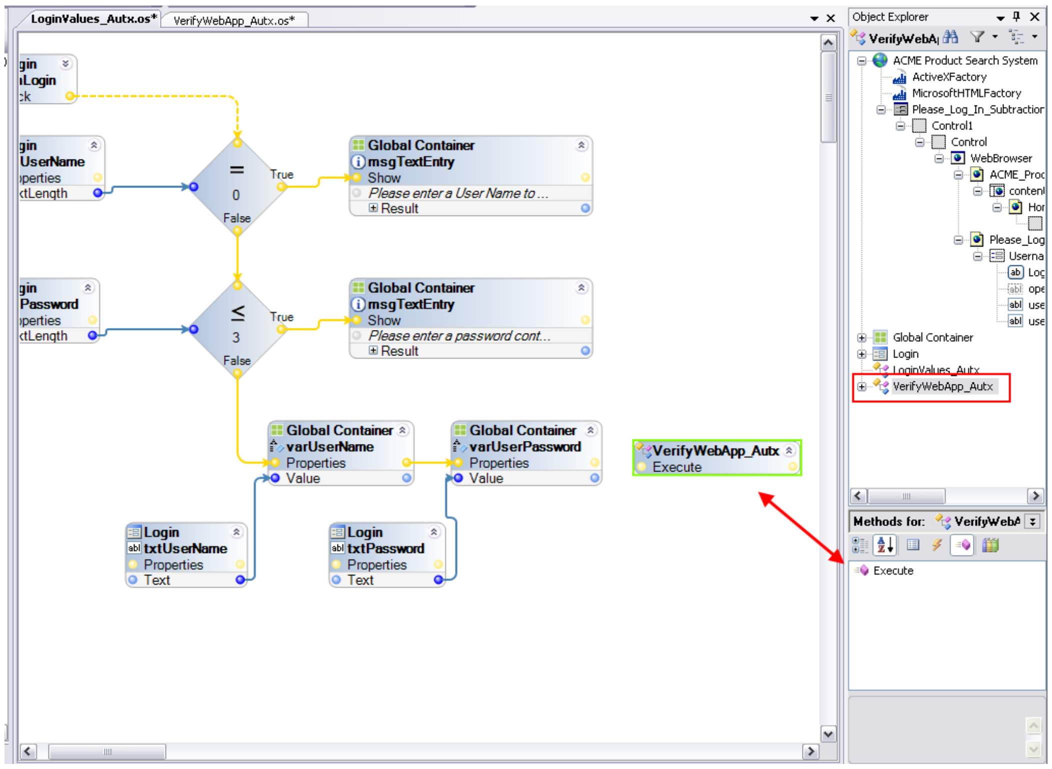
4. Connect the components on the **VerifyWebApp_Autx** automation as follows:

Source Component	Connect To	Target Component
Execute Entry Point	→	ACME Product Search System web application.IsRunning property execution input
ACME Product Search System web application.IsRunning property (True event)	→	ACME Product Search System web application.Please_ Log_In web page.IsCreated property execution input
ACME Product Search System web application.IsRunning property (False event)	→	ACME Product Search System web application.Start method execution input
ACME Product Search System web application.Please_ Log_In web page.IsCreated property (False event)	→	WebBrowser program.Navigate (1 parameter) method execution input
ACME Product Search System web application.StartPage property data output	→	WebBrowser program.Navigate (1 parameter) method data input
ACME Product Search System web application.Start method execution output	→	ACME Product Search System web application.Login button.WaitForCreate(<no parameters>) method execution input
WebBrowser program.Navigate (1 parameter) method execution output	→	ACME Product Search System web application.Login button.WaitForCreate(<no parameters>) method execution input

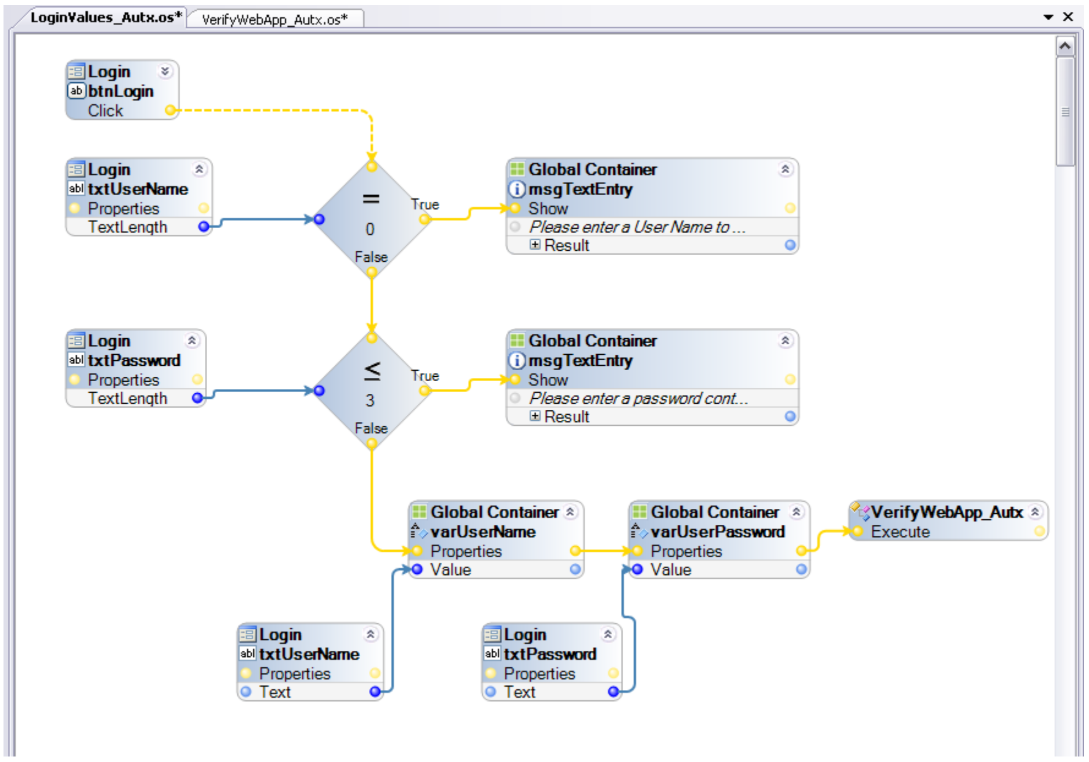
Your automation should look similar to the following:



5. Return to the **LoginValues_Autx** automation and add the **VerifyWebApp_Autx.Execute** method.



6. Connect the **VarUserPassword.Properties** execution output to the **VerifyWebApp_Autx.Execute** method execution input. This connection will start the adapter verification process after the text entries on the Login window have been verified. The **LoginValues_Autx** automation should look like the following:



7. Select **File | Save All**.

Test Login Automation - Part 2

Test your automation logic with the following scenarios:

- Run the solution and navigate the web browser to another page (such as www.google.com), and then enter login information in the **My Application Login** window and click the **Login** button. *The browser should automatically navigate back to the Web Certification site Login page.*

- Run the solution and wait for the web browser to display the login page. Close browser and then enter login information on the Login window and click the **Login** button. *The browser should automatically restart and display the Web Certification site Login page.*

Once you have tested the solution, stop the solution and return to OpenSpan Studio.

Add Login Automation Logic

In completing the previous automations, you have established the logic to pass the login values from the **Login** window to the **ACME Product Search System** web application. In passing the values the solution should:

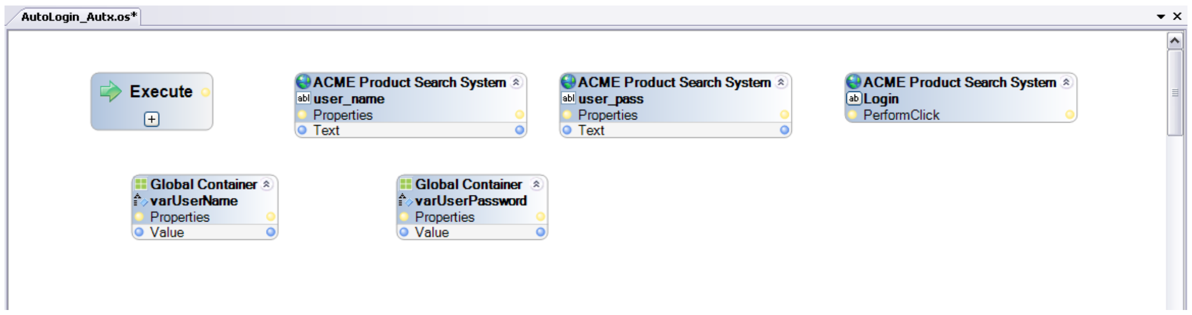
- Ensure that data links for the **Password** are sensitive so that they will not be captured in the log.
- Save the **Username** and **Password** in solution variables so that should the user need to re-login following a logout from the web application, the solution can send the values directly to the web application login page and not require the user to re-type or re-submit the information.
- Raise the appropriate event for the **Login** button.

1. Add a new automation to the solution: **AutoLogin_Autx**
2. Set the **ShowDesignCompNames** property to **True** for the **AutoLogin_Autx** automation.
3. Add the following components to the **AutoLogin_Autx** automation.

Component	Source Project Item	Property	Event	Method
Execute (entry point)	Right-click in design pane and select Add Entry Point from the context menu to add the Execute Entry Point.			
user_name textbox	ACME Product Search System.Please_Log_In webpage.	Text		
user_pass textbox	ACME Product Search System.Please_Log_In webpage.	Text		
Login button	ACME Product Search System.Please_Log_In webpage.			PerformClick
varUserName string variable	Global Container	Value		

Component	Source Project Item	Property	Event	Method
varUserPassword string variable	Global Container	Value		

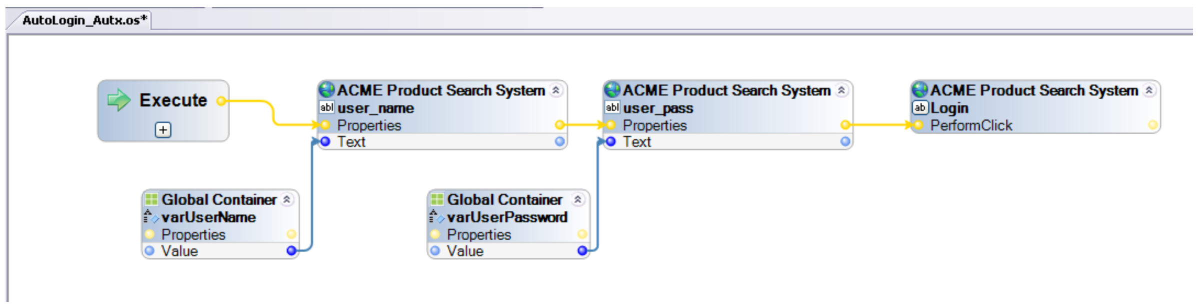
Your automation should appear as follows:



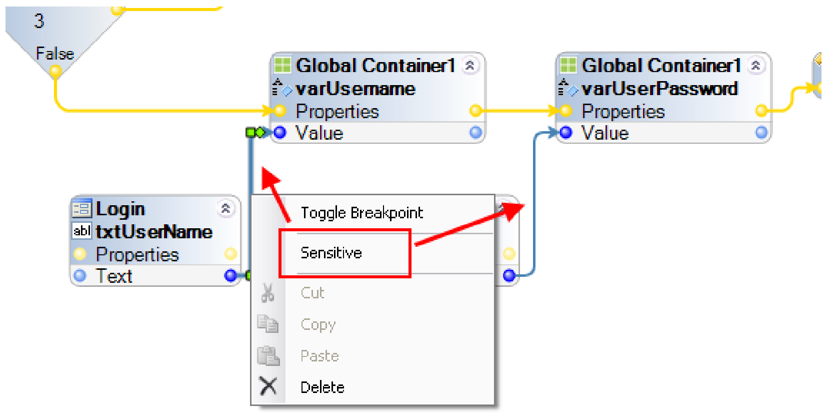
Connect the components on the **AUTX_AutoLogin** automation:

Source Component	Connect To	Target Component
Execute Entry Point output	→	ACME Product Search System web application.user_name textbox.Text execution input
Global Container.varUserName.Value property data output. Set the data link to Sensitive by right-clicking the data connector (blue line) and selecting **Sensitive** from the context menu.	→	ACME Product Search System web application.user_name textbox.Text data input
ACME Product Search System web application.user_name textbox.Text property execution output	→	ACME Product Search System web application.user_pass textbox.Text property execution input.
Global Container.varUserPassword.Value property data output. Set the data link to **Sensitive**	→	ACME Product Search System web application.user_pass textbox.Text property data input
ACME Product Search System web application.user_pass textbox.Text property execution output	→	ACME Product Search System web application.Login button.PerformClick method execution input

Your automation should look like this:



4. Select the **LoginValues_Autx** automation. Set the links between the username and password Design blocks to be sensitive.

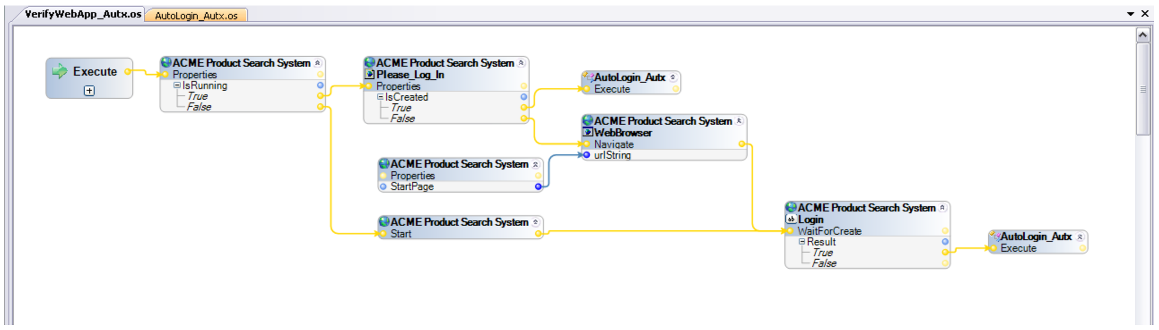


5. Return to the **VerifyWebApp_Autx** and add two connection blocks of the **AutoLogin_Autx.Execute** method.

6. Connect the components as follows:

   - **True** event from **ACME Product Search System** web application.**Login** button.**WaitForCreate** method to the execution input of the **AutoLogin_Autx.Execute** method.
   - **True** event from the **ACME Product Search System** web application.**Please_ Log_In** web page.**IsCreated** property to the **AutoLogin_Autx.Execute** method.

Your automation should similar to the following:



7. Select **File | Save All**.

To test your automation logic, run the solution and complete the **My Application Login** window fields then press the **Login** button on the window. You would expect that the data will be automatically copied to the web application and the login executed. However, the **Login** button on the Web Adapter Certification Course website remains disabled! To determine why this is the case you need to view the source HTML to determine how this button becomes enabled.

Exercise – Raising Web Control Events

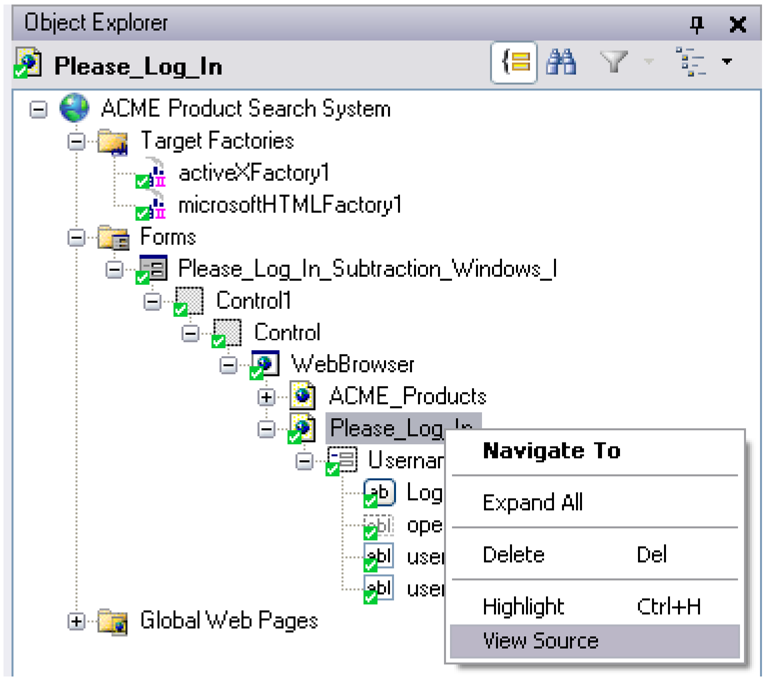
The login page for the ACME Product Search System uses a script to validate the password entry. The page was designed such that the password must be at least 4 characters before the **Login** button is enabled. While our automation sends text of four or more characters to the **Password** field, the process of validating the entry has not been automated and so the button does not become enabled.

To determine how the password text is validated and the button enabled, you need to view the source HTML for the page. Use the following steps to determine how to enable the Login button.

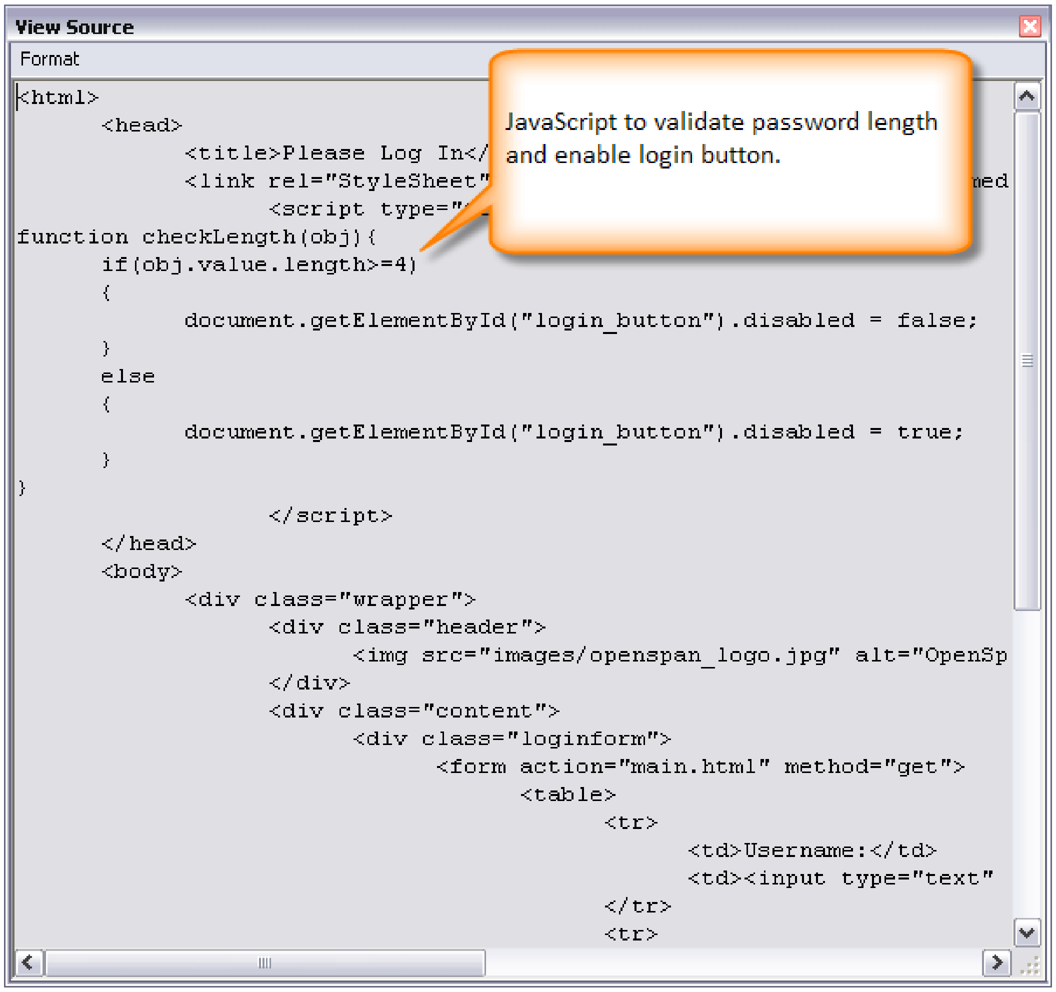
1. Stop the solution if it is currently running.

2. Open the **ACME Product Search System** web application and start the Interrogator.

3. Right-click the **Please_Log_In** page in the **Object Explorer** and select **View Source**.

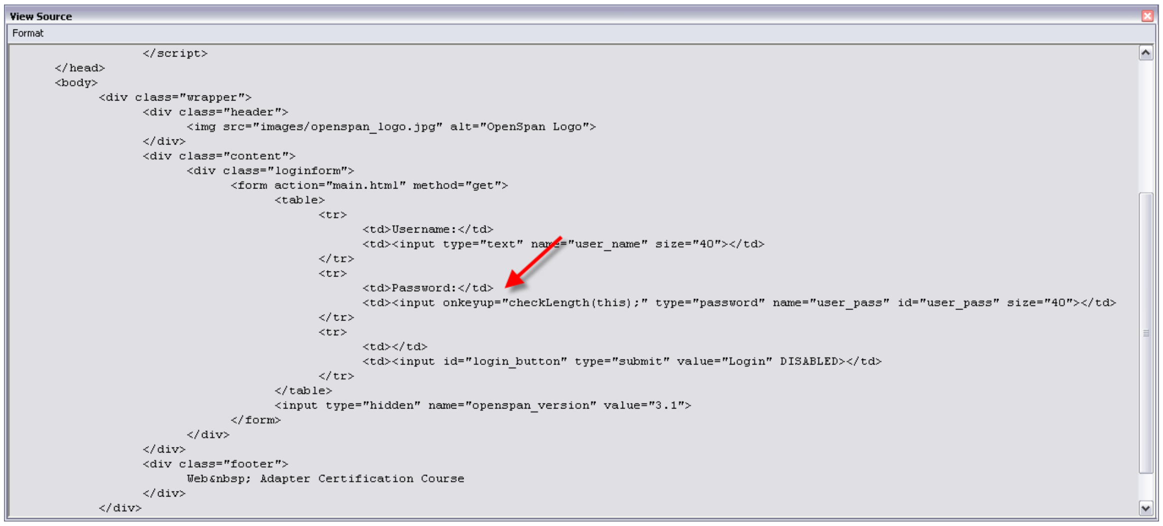


4. The HTML source for the page displays.



```
<html>
        <head>
                <title>Please Log In</
                <link rel="StyleSheet"                                       ned
                        <script type="
function checkLength(obj){
        if(obj.value.length>=4)
        {
                document.getElementById("login_button").disabled = false;
        }
        else
        {
                document.getElementById("login_button").disabled = true;
        }
}
                        </script>
        </head>
        <body>
                <div class="wrapper">
                        <div class="header">
                                <img src="images/openspan_logo.jpg" alt="OpenSp
                        </div>
                        <div class="content">
                                <div class="loginform">
                                        <form action="main.html" method="get">
                                                <table>
                                                        <tr>
                                                                <td>Username:</td>
                                                                <td><input type="text"
                                                        </tr>
                                                        <tr>
```

5. Search the text for reference to the Password:



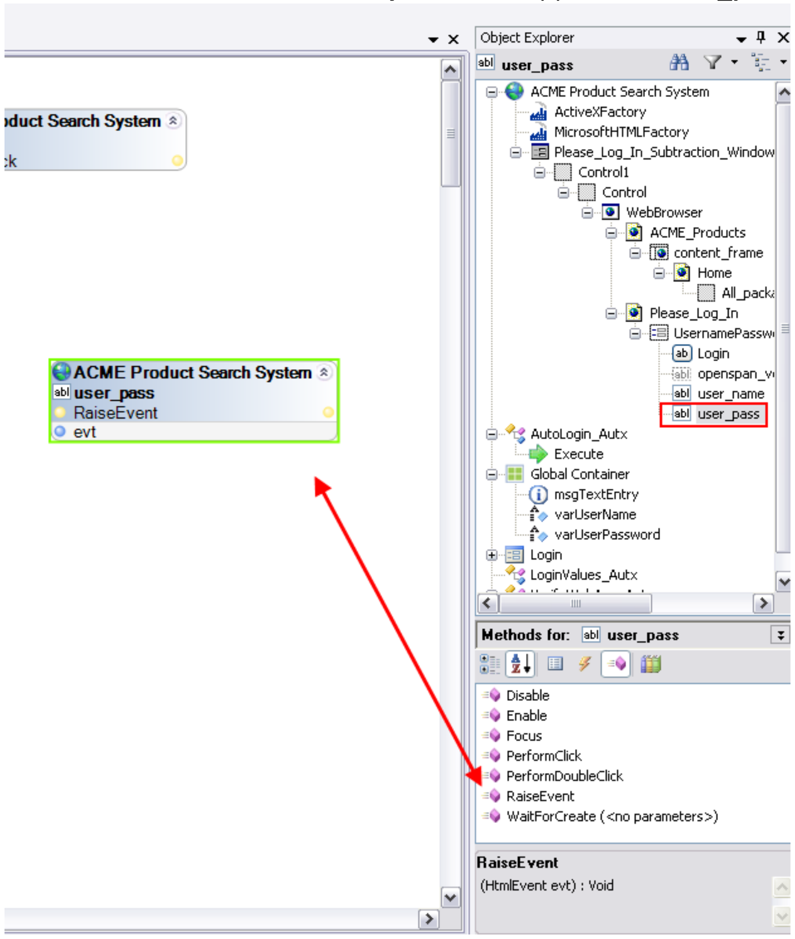
```
                        </script>
        </head>
        <body>
                <div class="wrapper">
                        <div class="header">
                                <img src="images/openspan_logo.jpg" alt="OpenSpan Logo">
                        </div>
                        <div class="content">
                                <div class="loginform">
                                        <form action="main.html" method="get">
                                                <table>
                                                        <tr>
                                                                <td>Username:</td>
                                                                <td><input type="text" name="user_name" size="40"></td>
                                                        </tr>
                                                        <tr>
                                                                <td>Password:</td>
                                                                <td><input onkeyup="checkLength(this);" type="password" name="user_pass" id="user_pass" size="40"></td>
                                                        </tr>
                                                        <tr>
                                                                <td></td>
                                                                <td><input id="login_button" type="submit" value="Login" DISABLED></td>
                                                        </tr>
                                                </table>
                                                <input type="hidden" name="openspan_version" value="3.1">
                                        </form>
                                </div>
                        </div>
                        <div class="footer">
                                Web&nbsp; Adapter Certification Course
                        </div>
                </div>
```

6. This text indicates that the **onkeyup** event triggers the **CheckLength** function. To automate the login process, your solution needs to raise the **onkeyup** event so that the **CheckLength** function can complete.

7. Close the **View Source** window and stop the Interrogator.

8. Return to the **AutoLogin_Autx** automation.

9. Add the **ACME Product Search System** web application.**user_pass** textbox.**RaiseEvent** method.

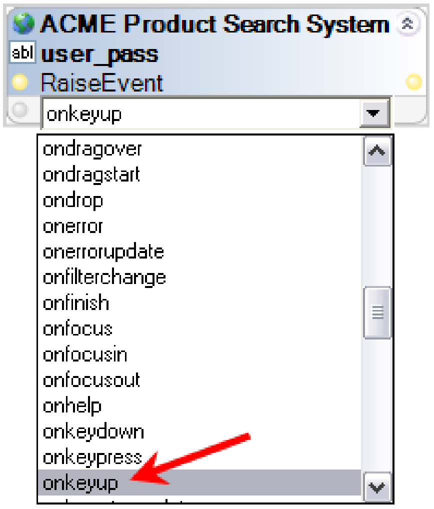


Note: If the RaiseEvent method does not appear in the lower pane of the Object Explorer for the **user_pass** textbox, then select the Configure Type icon to open the **TextBox Configuration** dialog. In the **TextBox Configuration** dialog, expand the **Methods** item in the tree view and select the **RaiseEvent** check box, and then click **OK** to save your selection.

10. Disconnect the input execution link from the **ACME Product Search System** web application.**Login** button.**PerformClick** method.

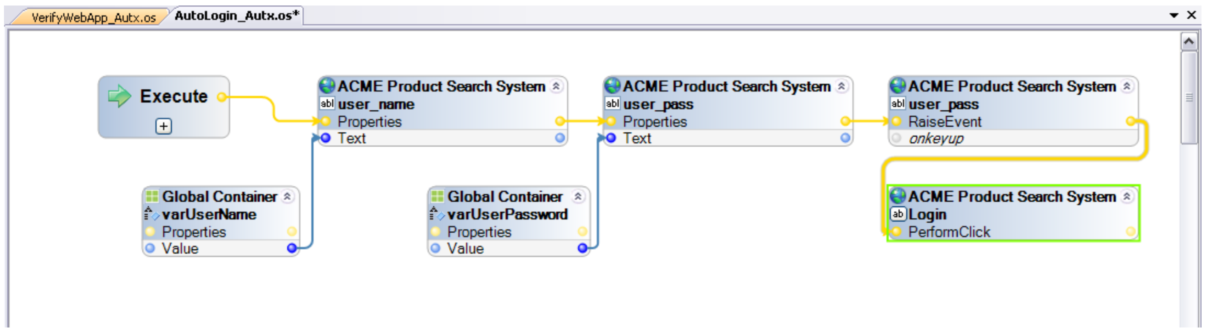
11. Connect the **ACME Product Search System** web application.**user_pass** textbox execution output to the execution input point of the **ACME Product Search System** web application.**user_pass** textbox.**RaiseEvent** method.

12. Click on the **evt** property of the **RaiseEvent** method and select the **onkeyup** event.



13. Connect the **RaiseEvent** output event to the **PerformClick** method of the Login Button.

Your automation should look like the following:



14. Select **File | Save All**.

Test Login Automation Part 3

Test your automation logic with the following scenarios:

- Run the solution and complete the **My Application Login** window fields. Then press the **Login** button on the window. *The data should automatically be copied to the web application and the login executed opening on the ACME Product Search System page.*
- Run the solution and complete the **My Application Login** window fields then press the **Login** button on the window. Once you are logged in, close the web application by closing Internet Explorer. Then click the **Login** button again on the **My Application Login** window. *Internet Explorer should re-launch the web application and return to the login page; next the data should be automatically copied to the web application and the login executed resulting on the ACME Product Search System page.*

15. Stop the solution.

CHAPTER 5: WEB APPLICATION NAVIGATION

A common feature of a web application is the use of a navigation bar and the use of hyperlinks. Users navigate the application by clicking the hyperlinks or using the controls on the navigation bar. This chapter describes how to incorporate web links in your OpenSpan solutions. Topics covered include:

- Using the Global Web Page Interrogation option
- Using the PerformClick method on links
- Working with JavaScript menus
- Navigating a New Browser instance

Interrogating Links

On the OpenSpan training website, each page beyond the **Login** page contains links at the top of the page for navigation. To use these links in your automation you need to interrogate them such that the links match no matter which page the user is on when using the website. One way to ensure that the links match each page is to interrogate each page and interrogate the links. However, the links have been placed in a frame that enables us to use the **Create Global Web Page** option when interrogating the links. This way, you only have to interrogate the links once, and they will match no matter which page you are on within the training website.

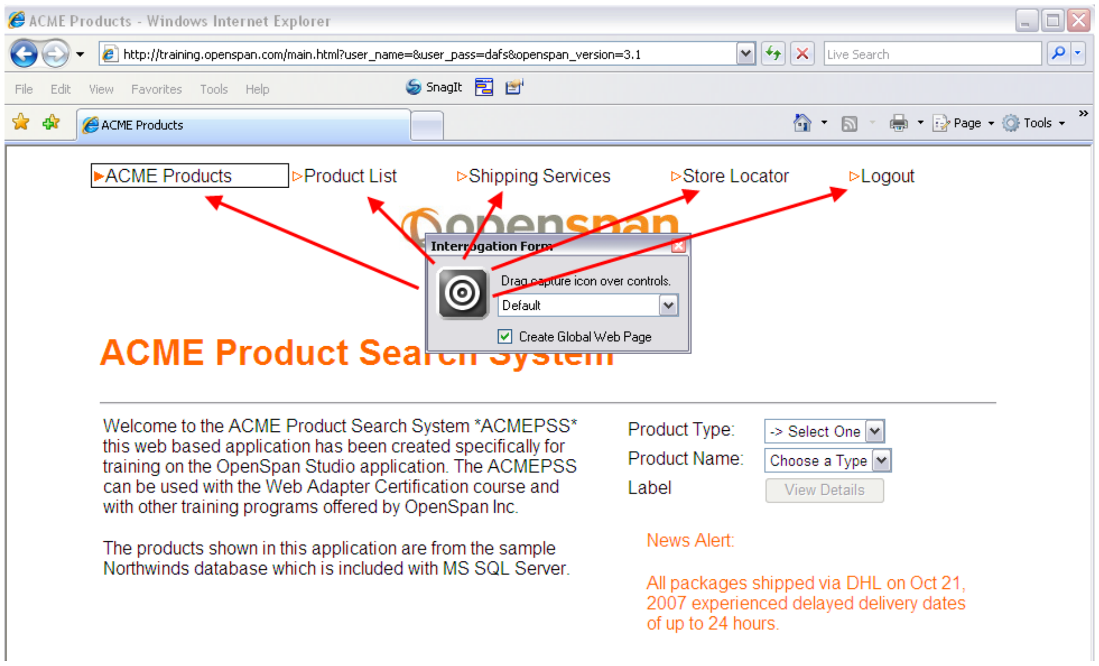
Exercise – Interrogate Links

1. Return to the Web Adapter Certification solution and open the **ACME Product Search System** web application.
2. Start the Interrogator. Complete the **Login** page fields and select the **Login** button.

Create Global Web Page for Links

3. With the **Interrogation Form** dialog set to **Default,** select the **Create Global Web Page** check box and then interrogate the following links (menu options):

   - ACME Products
   - Product List
   - Shipping Services
   - Store Locator
   - Logout

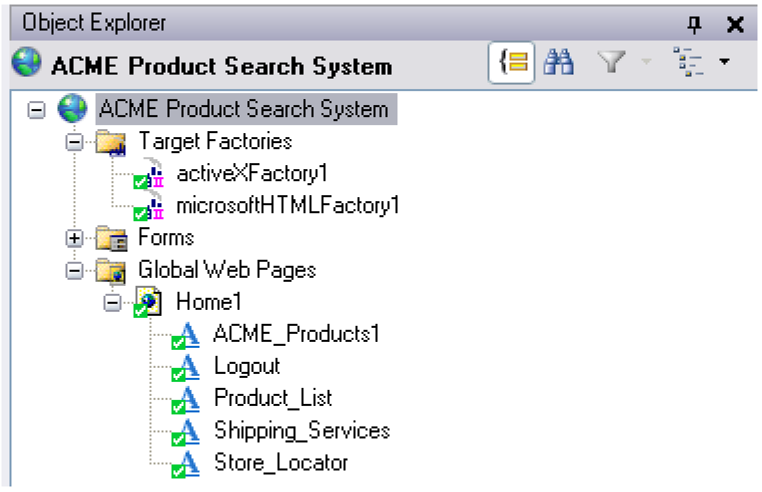

ACME Products - Windows Internet Explorer
http://training.openspan.com/main.html?user_name=&user_pass=dafs&openspan_version=3.1
File Edit View Favorites Tools Help
ACME Products
▶ACME Products
▷Product List
▷Shipping Services
▷Store Locator
▷Logout
Interrogation Form
Drag capture icon over controls.
Default
Create Global Web Page
ACME Product Search System
Product Type:
-> Select One
Product Name:
Choose a Type
Label
View Details
News Alert:

Welcome to the ACME Product Search System *ACMEPSS* this web based application has been created specifically for training on the OpenSpan Studio application. The ACMEPSS can be used with the Web Adapter Certification course and with other training programs offered by OpenSpan Inc.

The products shown in this application are from the sample Northwinds database which is included with MS SQL Server.

All packages shipped via DHL on Oct 21, 2007 experienced delayed delivery dates of up to 24 hours.

Your Object Explorer should look like the following:



4. With the interrogator running, navigate to the other pages of the site by clicking the navigation links (menu options). No matter which page displays (except the **Login** page), the links should remain matched.

Automate Link Selection

Having interrogated the links using the **Create Global Web Page** option, any time the page containing the links appears, it will be matched. You can use the **PerformClick** method for the links to automate navigation. Continue the exercise by adding an automation to move directly to the **Beverages** page after login.

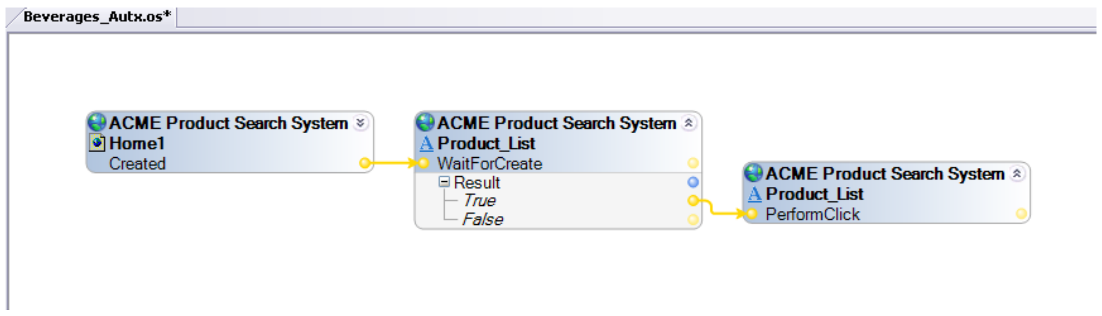
1. Stop the interrogator and save the solution.
2. Add a new automation to your solution named **Beverages_Autx**.
3. Set the **ShowDesignCompNames** property to **True**.
4. Add the following components to the automation:

Object	Source Project Item	Property	Event	Method
Home1 page	ACME Product Search System		Created	
Product_List link (menu)	ACME Product Search System			WaitForCreate Perform Click

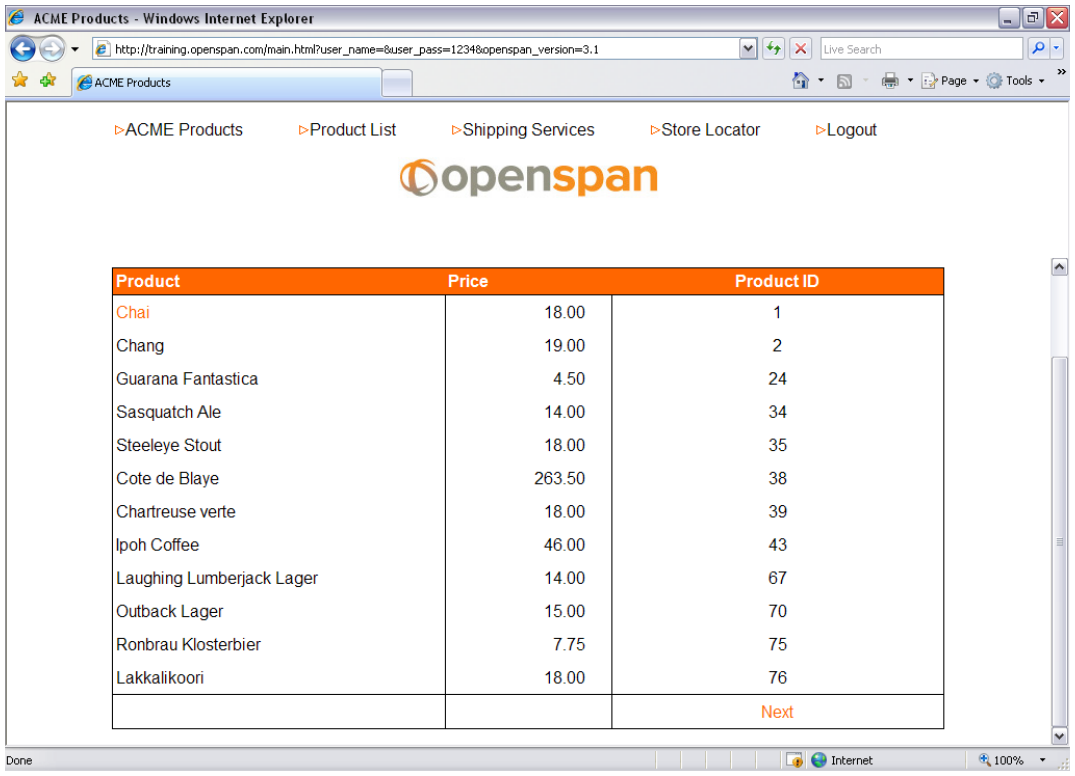
5. Connect the components as follows:

Source Component	Connect To	Target Component
ACME Product Search System web application Home1 page.Created execution output	→	ACME Product Search System web application Product_List link.WaitForCreate method execution input
ACME Product Search System web application Product_List link.WaitForCreate method True execution output	→	ACME Product Search System web application Product_List link.Perform Click method execution input

The **Beverages_Autx** automation should look like the following:



6. Save and run the solution. Once you login, the browser should automatically be navigated to the **Product List – Beverages** page:

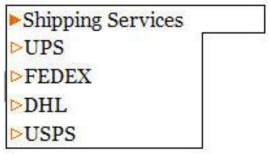

ACME Products - Windows Internet Explorer
http://training.openspan.com/main.html?user_name=&user_pass=1234&openspan_version=3.1
Live Search
ACME Products
Page
Tools
▷ACME Products
▷Product List
▷Shipping Services
▷Store Locator
▷Logout
openspan
Done
Internet
100%

Product	Price	Product ID
Chai	18.00	1
Chang	19.00	2
Guarana Fantastica	4.50	24
Sasquatch Ale	14.00	34
Steeleye Stout	18.00	35
Cote de Blaye	263.50	38
Chartreuse verte	18.00	39
Ipoh Coffee	46.00	43
Laughing Lumberjack Lager	14.00	67
Outback Lager	15.00	70
Ronbrau Klosterbier	7.75	75
Lakkalikoori	18.00	76
		Next

7. Stop the solution.

Navigating Links

In some cases, navigation links are simple HTML links to specific pages or frames. However, some web applications use JavaScript menus such as the one presented in the ACME Products Search System **Shipping Services** link. When you hover over the **Shipping Services** menu a submenu of shippers displays.

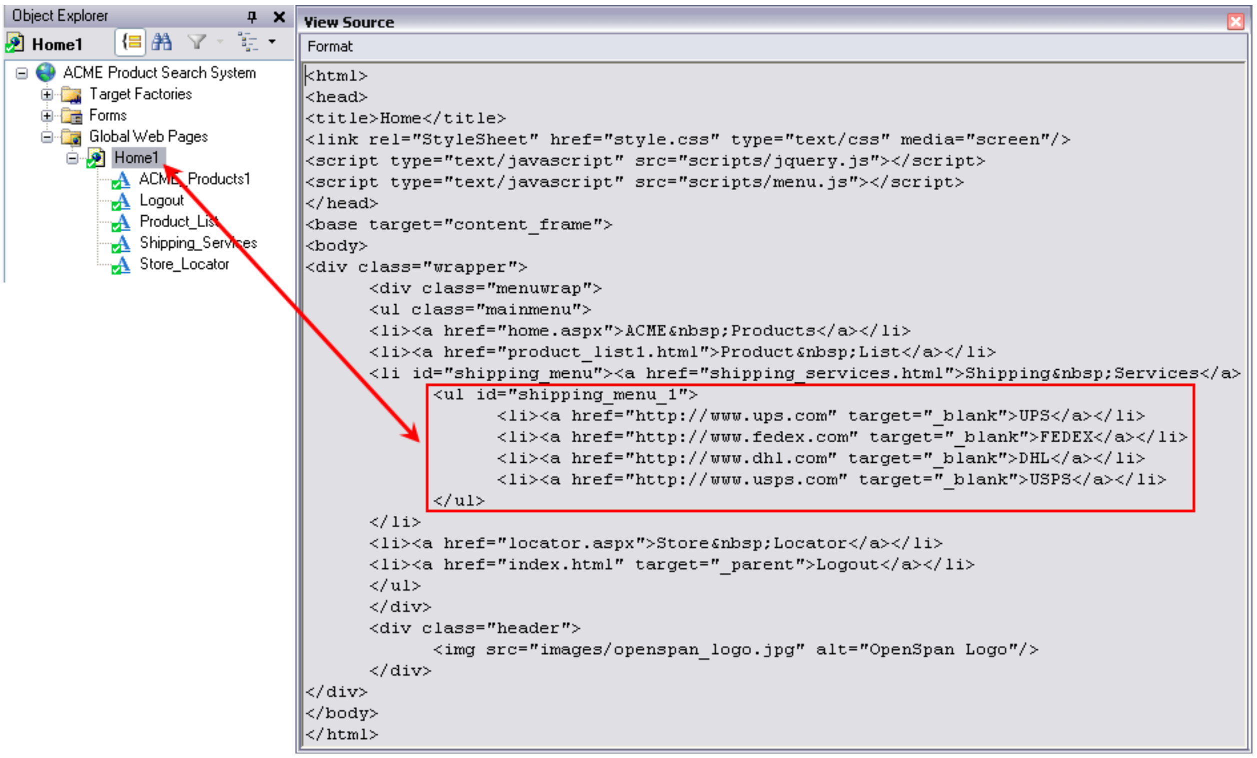


You cannot interrogate these links directly. In order to automate selection of this type of menu option, you need to:

- Determine the script used by the menu link
- View the script to determine the menu option actions

In the following exercise, you will get practice working with JavaScript menus.

1. Open the **Web Adapter Certification** solution.
2. Start interrogation on the **ACME Product Search System** web application.
3. Login to the application.
4. Right-click the **Home1** page in the **Object Explorer** and select **View Source**. The HTML source for the page displays:

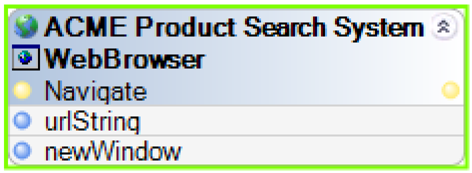


```
<html>
<head>
<title>Home</title>
<link rel="StyleSheet" href="style.css" type="text/css" media="screen"/>
<script type="text/javascript" src="scripts/jquery.js"></script>
<script type="text/javascript" src="scripts/menu.js"></script>
</head>
<base target="content_frame">
<body>
<div class="wrapper">
      <div class="menuwrap">
      <ul class="mainmenu">
      <li><a href="home.aspx">ACME&nbsp;Products</a></li>
      <li><a href="product_list1.html">Product&nbsp;List</a></li>
      <li id="shipping_menu"><a href="shipping_services.html">Shipping&nbsp;Services</a>
            <ul id="shipping_menu_1">
                  <li><a href="http://www.ups.com" target="_blank">UPS</a></li>
                  <li><a href="http://www.fedex.com" target="_blank">FEDEX</a></li>
                  <li><a href="http://www.dhl.com" target="_blank">DHL</a></li>
                  <li><a href="http://www.usps.com" target="_blank">USPS</a></li>
            </ul>
      </li>
      <li><a href="locator.aspx">Store&nbsp;Locator</a></li>
      <li><a href="index.html" target="_parent">Logout</a></li>
      </ul>
      </div>
      <div class="header">
            <img src="images/openspan_logo.jpg" alt="OpenSpan Logo"/>
      </div>
</div>
</body>
</html>
```

The source indicates that when a menu option is selected, the corresponding link is opened in a new browser. You can accomplish the same logic by using **Browser.Navigate** method with 2 parameters: **String urlString**, **Boolean newWindow**. Use the following steps to automate opening the **FEDEX** link.

5. Close the **View Source** window, stop the Interrogator, and save the solution.

6. Return to the **Beverages_Autx** automation and add the **ACME Product Search System** web application.**WebBrowser**.**Navigate** (2 parameters) method.



7. Set the **URL String** to www.fedex.com.

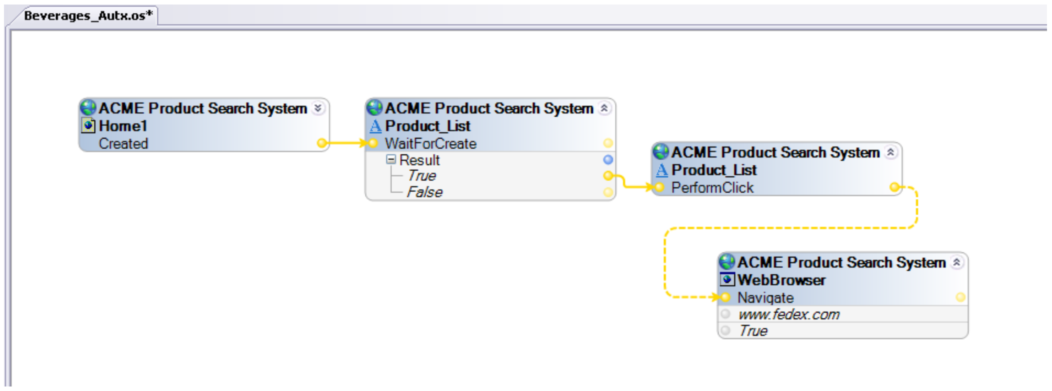
8. Set the **newWindow** parameter to **True**.

9. Connect the **ACME Product Search System** web application.**Product_List** link.**Perform Click** method execution output to the **ACME Product Search System** web application.**WebBrowser**.**Navigate** (2 parameters) method execution input.

   Note: If the **Navigate (2 parameters)** method does not appear in the lower pane of the Object Explorer for the **WebBrowser**, then select the Configure Type icon to open the **WebBrowser Configuration** dialog. In the **WebBrowser Configuration** dialog, expand the **Methods** item in the tree view and select the **Navigate (2 parameters)** check box, and then click **OK** to save your selection.

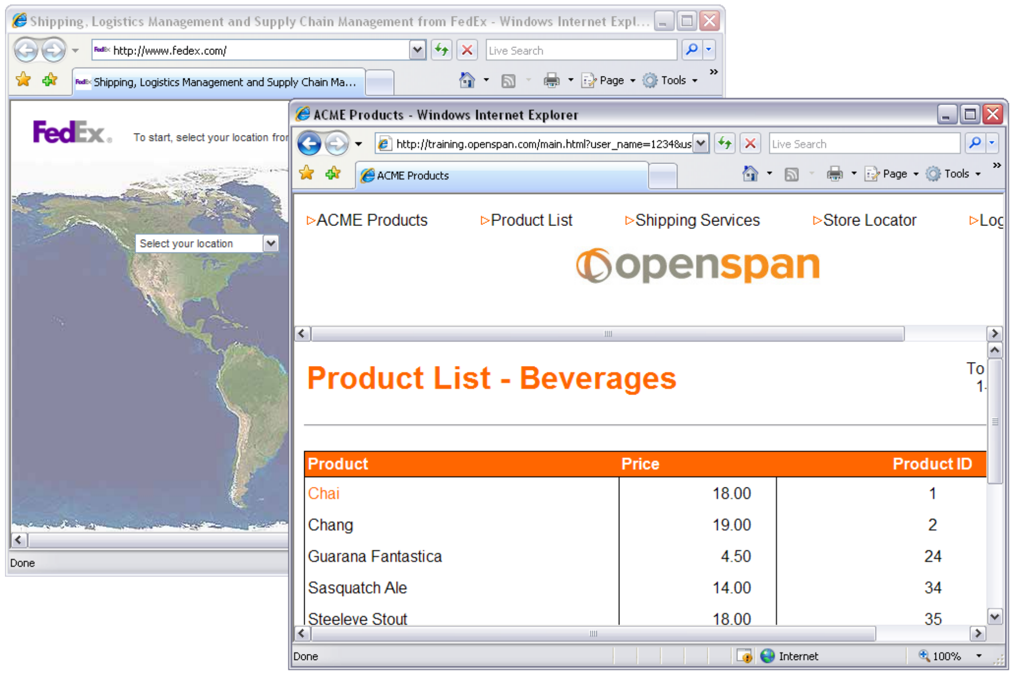
10. Set the event link between the **PerformClick** method and the **Navigate** method to **Asynchronous** (so that the ACME Product Search System is not locked while the new browser is launched).

    Your automation should look like the following:



11. Save and run the solution. After you login, two browser windows should open.

    Note: Disable your Pop-up Blocker before running the solution.


Shipping, Logistics Management and Supply Chain Management from FedEx - Windows Internet Expl...
http://www.fedex.com/
Live Search
Shipping, Logistics Management and Supply Chain Ma...
Page
Tools
FedEx
To start, select your location fro
Select your location
Done
ACME Products - Windows Internet Explorer
http://training.openspan.com/main.html?user_name=1234&us
Live Search
ACME Products
▷ACME Products
▷Product List
▷Shipping Services
▷Store Locator
▷Log
openspan
To
1
Done
Internet
100%

Product List - Beverages

Product	Price	Product ID
Chai	18.00	1
Chang	19.00	2
Guarana Fantastica	4.50	24
Sasquatch Ale	14.00	34
Steeleye Stout	18.00	35

Final Exercise - Completing the Auto-Login Process

A few changes to the solution will make the user experience even easier.

- Have the Acme Product Search System launch on demand when called by the **My Application Login** window. *Use the StartOnProjectStart property.*

- Once the login process is complete, hide the **My Application Login** window since it is no longer required. *Use the Hide method for the window at the end of the Autx_AutoLogin logic.*

Completed Solution

A completed *Web Adapter Certification* solution in the form of a deployment file can be found in the **Extras\Advanced** sub-folder of the **OpenSpan Studio** program folder. The deployment file name is *Web Adapter Certification.openspan*.

This page intentionally left blank.