

Lab 6: Calling Server Applications

Lab Overview

Objectives

After completing this lab, you will be able to:

- ♦ Create a Form and set its ACTION and METHOD properties.
- ♦ Use the FormDump server extension as a testing mechanism for a Form.
- ♦ Use the ReDir server extension to jump to HTTP sites listed in a list box.
- ♦ Use the OLEISAPI server extension to call an ActiveX component running on the Internet server.

Scenario

In previous labs, you have created Web pages that have active components such as VBScript code and controls. However, any information entered into these controls is lost when the user moves to the next page.

In this lab, you will add calls to server applications. You will also pass information from controls on your Web pages to the server where it is processed.

Lab Setup

In this lab, you add functionality to three of the Main Street Market Web pages: CustData.htm, MSMBody.htm, and Products.htm. You can either continue to edit the files you worked with in Lab 5, or use the files provided in the \Labs\Lab06 directory.

The ISAPI applications you call are provided as compiled DLLs in the \Labs\Lab06 subdirectories.

If you need help, the solutions for all of the exercises in this lab can be found in the \Labs\Lab06\Solution directory.

This demonstration shows the solution to this lab.



Estimated time to complete this lab: **60 minutes**.

Exercises

The following exercises provide practice working with the concepts and techniques covered in this chapter.

Exercise 1: Calling FormDump.dll

In this exercise, you will add a Submit button to the Customer Data Web page and call the FormDump server extension.

Exercise 2: Calling ReDir.dll

In this exercise, you will add a list of related Web sites to the Main Street Market home page and use the ReDir server extension to jump to them.

Exercise 3: Calling an ActiveX Component

In this exercise, you will pass information from the Products table to an ActiveX component on the Internet server.

Exercise 1: Calling FormDump.dll

In Lab 4, you created a Web page for obtaining information about Main Street Market customers. In Lab 5, you wrote VBScript code to verify that the controls were completely filled. In this lab, you will configure the form to send the customer information to the Internet Server.

➤ Copy FormDump.dll to the \Scripts directory

- ◆ Copy the file, FormDump.dll from the \Labs\Lab06\FormDump directory to the \Scripts directory of your Internet Server.

Note The source files for the FormDump server application are also in the \Labs\Lab06\FormDump directory.

➤ Set properties of the Form, and add a Submit button

1. Open the Customer Data page, CustData.htm, in Notepad.
2. Locate the <FORM> tag that was added by Internet Assistant for Word, and add the following properties:
 - a. Set ACTION to /Scripts/FormDump.dll
 - b. Set METHOD to POST.

If you are unsure of the HTML syntax, look at the solution:

```
<FORM ACTION="/scripts/formdump.dll" METHOD="POST">
```

3. Add a submit button to the form (TYPE="SUBMIT").
4. Save your changes to the page.
5. Open the Customer Data page in Internet Explorer, enter data into the controls on the page, and select the Submit button. Notice the values of the controls sent to the server.

➤ Add HTML hidden controls for the Phone and Fax numbers

1. Add two HTML hidden controls to the form to hold the phone and fax numbers entered into the Phone and Fax masked edit controls:
 - a. Set TYPE to HIDDEN.
 - b. Set NAME to PhoneNum and FaxNum, respectively.
 - c. Set VALUE to an empty string.

2. Add a NAME parameter to the <FORM> tag:

```
<FORM NAME="DataForm" ACTION="/Scripts/FormDump.dll" METHOD="POST">
```

3. Create an OnSubmit event procedure for the DataForm form:

- a. Set the VALUE of the PhoneNum hidden control to the Text property of the Phone masked edit control.
- b. Set the VALUE of the FaxNum hidden control to the Text property of the Fax masked edit control.

If you are unsure of the implementation of this event procedure, look at the solution:

```
Sub DataForm_OnSubmit()  
    Dim x  
    Set x = Document.Forms(0)  
    'save values of the Phone and Fax masked edit controls  
    x.PhoneNum.Value = x.Phone.Text  
    x.FaxNum.Value = x.Fax.Text  
End Sub
```

4. Save your changes to the page.
5. Open the Customer Data page in Internet Explorer, enter data into the controls on the page, and select the Submit button. Notice the values of the PhoneNum and FaxNum controls sent to the server.

➤ **Try to stop the form submission from a submit button**

1. Add an OnClick event procedure to the Submit button.
2. In the procedure, call the Verify procedure you wrote in Lab 5. This procedure tests the validity of the data entered in the controls. If the value returned by Verify is False, run the **Exit Sub** statement.
3. Save your changes to the page.
4. Open the Customer Data page in Internet Explorer, leave one or more of the controls empty, and select the Submit button. Notice that even if you **Exit Sub**, the form is still submitted.

➤ **Use the form's Submit method**

1. Edit the Test button's OnClick event procedure:
 - a. If all controls are valid, invoke the Submit method of the form with `DataForm.Submit`.

Note By explicitly calling the Submit method of a form, instead of implicitly calling it through a button, you can dynamically decide when to submit data to the server.

2. Save your changes to the page.
3. Open the Customer Data page in Internet Explorer, leave one or more of the controls empty, and select the Test button. Notice that the form is only submitted if all controls are filled in.

Exercise 2: Calling ReDir.dll

Web sites commonly include links to other related sites. If the number of related sites is large, this could take up considerable space on a page. In this exercise, you will add to the Main Street Market home page a form with a list of related sites for customers to visit.

➤ **Copy ReDir.dll to the \Scripts directory**

- ◆ Copy the file ReDir.dll from the \Labs\Lab06\ReDir directory to the \Scripts directory of your Internet Server.

Note The source files for the ReDir server application are also in the \Labs\Lab06\ReDir directory.

➤ **Supply a list box of other Web pages**

Your finished form will look similar to the following:

The image shows a web form with a list box containing the text 'Recipe of the Week' and a small downward arrow on the right. To the right of the list box is a rectangular button with the text 'GO' centered on it.

1. Edit the body frame of the Main Street Market home page, MSMBody.htm, with Internet Assistant for Word.
2. Add a list box of related Web sites.
 - a. Set the NAME to Target.
 - b. Add entries for some of your favorite sites.
 - c. Set the VALUE of each entry you add to the full URL of the HTTP site.

Some possible Web sites include:

- Microsoft's Julia Child CD Home Page: <http://www.microsoft.com/MasterChef/>
- The Julia Child CD Recipe of the week: <http://www.microsoft.com/MasterChef/Recipe/Recipe.htm>
- Microsoft's Wine Guide CD Home page: <http://www.microsoft.com/wineguide/>
- Yahoo's Cooking and Food Page: <http://www.yahoo.com/entertainment/food/>

3. Add a submit button to the form. Fill in the Submit Button form field dialog box with the following information:
 - a. Leave the NAME of the button blank.
 - b. Set the Button Label to GO.
 - c. Set the ACTION to /Scripts/ReDir.dll.
 - d. Set the METHOD to GET.
4. Save your changes to the page.

➤ **Test the list box**

- ◆ Open the Main Street Market home page in Internet Explorer, select an entry in the list box, and click Go.

Exercise 3: Calling an ActiveX Component

In this exercise, you will add to the Products Web page a button that calls an ActiveX component on the Internet server. The ActiveX component takes the information in the Products table, processes it, and returns a confirmation to the user.

The ActiveX component is written in Visual Basic 4.0. The code for the Visual Basic project is in the \Labs\Lab06\OLESVr directory.

➤ **Copy OLEISAPI.dll to the \Scripts directory**

- ◆ Copy OLEISAPI.dll from the \Labs\Lab06\OLEISAPI directory to the \Scripts directory of your Internet server.

Note The source files for the OLEISAPI server application are also in the \Labs\Lab06\OLEISAPI directory.

➤ **Register the ActiveX component on the Internet server**

1. Copy the \Labs\Lab06\OLESVr\Setup directory and its contents to any location on your Internet server.
2. On the Internet server, run Setup.exe in the Setup directory.

➤ **Set permissions on the Internet server to run the ActiveX component**

With OLEISAPI and Windows NT 4.0, the IIS service tries to create the referenced ActiveX component using the permissions of the Internet Server Anonymous Logon account. However, by default, only Administrator and System have Access and Launch permissions for OLE objects. Therefore, to make OLEISAPI work on the Internet server, you need to add the Internet Server Anonymous Logon account to the permissions list.

1. Determine the name of the Internet Server Anonymous Logon account:
 - a. From the Microsoft Internet Server folder, start the Internet Service Manager, Inetmgr.exe.
 - b. Select the WWW service and then click Service Properties from the Properties menu.
 - c. Note the Username of the Internet Anonymous Logon, you will need this information to complete step 7 below.
 - d. Close the Internet Service Manager.
2. Run DCOMCnfg.exe from the C:\WinNT\System32 directory.
3. Click the Default Security tab.
4. In the Default Access Permissions section, click Edit Default.
5. In the Registry Value Permissions dialog, click Add.
6. In the Add Users and Groups dialog, click Search.

7. In the Find Account dialog:
 - a. Enter the name of the Internet Anonymous Logon account in the Find User or Group text box.
 - b. Click the Search Only In option.
 - c. Select your machine from the list of domains. It should be the first entry in the list.
 - d. Click Search.
 - e. Click Add to add the account and close the Find Account dialog.
8. Click OK to close the Add Users and Groups dialog.
9. Click OK to close the Registry Value Permissions dialog.
10. In the Default Launch Permissions section, click Edit Default and repeat steps 5 through 9.
11. Restart your computer.

➤ **Send Product information to the ActiveX component**

1. Open the Products Web page, Products.htm, with Notepad.
2. Add a generic button (TYPE="BUTTON") after the table of products:
 - a. Set the NAME to "Send."
 - b. Set the VALUE to "Send Order."
3. Create an OnClick event procedure for the Send button:
 - a. Read the values of all Quantity text boxes and create a list of parameters as ProductID=Quantity pairs. Separate each pair with an ampersand (&).
For example:
`"60=" & Quantity1.Value & "&"`
 - b. Add the Name=Value pair for the Send button, because the ActiveX component is expecting it:
`Send=Send+Order`
 - c. Append the parameters to a call to the OLEISAPI application and the ActiveX component. The name of the ActiveX component is Orders, the object is named **Products**, and the method is **Post**:
`/Scripts/OLEISAPI.DLL/Orders.Products.Post?<parameter string>`
 - d. Jump to the location you just created by setting Location.HRef to your string.

If you are unsure of how to implement this Sub procedure, look at the solution code:

```
Sub Send_OnClick()
    Dim submitStr
    submitStr = "60=" & Quantity1.Value & _
        "&58=" & Quantity2.Value & _
        "&65=" & Quantity3.Value & _
        "&70=" & Quantity4.Value & _
        "&55=" & Quantity5.Value & _
        "&21=" & Quantity6.Value & _
        "Send=Send+Order"
    submitStr = "/Scripts/OLEISAPI.DLL/Orders.Products.Post?" & _
        submitStr
    Location.HRef = submitStr
End Sub
```

4. Save your changes to the Products Web page.

➤ **Test the Send button**

- ◆ Open the Products Web page in Internet Explorer, add some values to the Quantity text boxes, and click Send.

The returned Web page should look similar to the following:

Thank you for your order:

| | | |
|------------|---------------------|--------|
| 3 | Camembert Pierrot | 34.00 |
| 2 | Outback lager | 15.00 |
| 1 | Sir Rodney's Scones | 10.00 |
| Sub Total: | | 142.00 |
| Tax: | | 11.71 |
| Total: | | 153.71 |

➤ **View the source code of the ActiveX component**

- ◆ If you have Visual Basic 4.0 on your computer, open the Orders.vbp project in the \Labs\Lab06\OLESVr directory. Look at the **Post** method of the **Products** class.