Lab 8: Using the Internet Explorer Object Model

Lab Overview

Objectives

After completing this lab, you will be able to:

- Start Internet Explorer from a Visual Basic application.
- Move to a page on the Web in the Internet Explorer by using Automation (formerly OLE Automation).
- Create a new Visual Basic form that uses the WebBrowser control to display Web sites.

Scenario

This lab adds functionality to a Visual Basic application called the "Main Street Market Sales Lead tool." Managers of Main Street Market use this tool to communicate sales leads to their salespeople. Each morning, a salesperson retrieves the most recent list of leads, via e-mail or some other mechanism, runs the Sales Lead tool, and contacts the customers that have been assigned to him or her.

Historically, this tool has only given phone numbers to the salespeople of Main Street Market. In this lab, you will extend the tool to include a Web site address for customers that have a Web site. This way, the salespeople of Main Street Market can research the customer before contacting them.

Lab Setup

This lab extends a Visual Basic application called the "Main Street Market Sales Lead Tool." You will edit the Visual Basic project, Leads.vbp, which is in the \Labs\Lab08 directory.

If you need help with this lab, solution files are in the \Labs\Lab08\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: Automating the Internet Explorer

In this exercise, you will use Automation (formerly OLE Automation) to start the Internet Explorer and display a Web page.

Exercise 2: Using the WebBrowser Control

In this exercise, you will add the WebBrowser control to a Visual Basic form to add Web browsing capabilities to your application.

Exercise 1: Automating the Internet Explorer

The Internet is a great mechanism for gathering information about companies. The Main Street Market sales people have found it invaluable for learning about their customers before trying to sell them products.

In this exercise you will enable users of the Sales Lead Tool to jump directly to an Internet site of a customer.

➢ Run the Sales Lead Tool

1. Open the Visual Basic project, Leads.vbp, in the \Labs\Lab08 directory, and run it.
2. Select Read Leads List from the File menu. This reads a file, Leads.mlt, from the local directory and displays the information in the lstLeads list view control.
3. View the entries in the File menu:

   Send E-mail won't be implemented in this lab.

   Open Web Site in Internet Explorer will open the Web site for the selected customer in a new instance of the Internet Explorer.

   Open Web Site in VB will open the Web site for the selected customer in a different form of the Sales Lead Tool.
4. Right-click on an entry in list, and look at the entries in the pop-up menu:

   Send E-mail, Open Web Site in Internet Explorer, and Open Web Site in VB have the same functionality as the same items on the File menu.

➢ Implement the Open Web Site in Internet Explorer menu item

1. Open the code window for the frmLeads form and locate the OpenWebInIE Sub procedure.

   This procedure is called from both of the Open Web Site in Internet Explorer menu items.
2. Read the Web site address from the selected customer in the lstLeads ListView control. The Web site address is SubItem 5:

   ```
   lstLeads.SelectedItem.SubItems(5)
   ```
3. Start a new instance of Internet Explorer.
4. Open the Web site for the selected customer by calling the **Navigate** method of the Internet Explorer.

Note The Web sites in the file Leads.mlt are fictional. You will want to substitute actual Web sites before testing. You can edit the file, Leads.mlt, with any ASCII text editor, such as Notepad.

---

5. Make Internet Explorer visible by setting the **Visible** property to True.
6. Save your changes.

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

```
Private Sub OpenWebInIE()
    Dim strWeb As String
    Dim browser As InternetExplorer
    'read the web site address from the selected list entry
    strWeb = lstLeads.SelectedItem.SubItems(5)

    'start up IE
    Set browser = CreateObject("InternetExplorer.Application")
    browser.Navigate strWeb
    browser.Visible = True
End Sub
```

➢ **Test the Open Web Site in Internet Explorer menu item**

1. Run the Sales Lead Tool.
2. Choose Read Leads List from the File menu.
3. Select a customer in the list.
4. Choose Open Web in Internet Explorer from the File menu or from the pop-up menu.

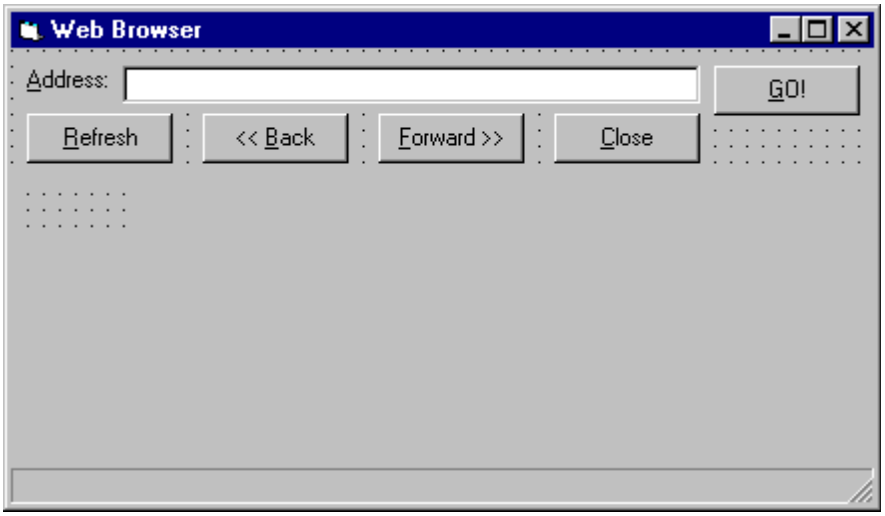
Exercise 2: Using the WebBrowser Control

The Internet Explorer is also an ActiveX control, called the WebBrowser, that you can include in your own application. Using the WebBrowser control gives you more control over exactly what functionality you provide your users.

In this exercise, you will add a form to the Main Street Market Sales Lead Tool that contains the WebBrowser control. This allows users to view Web sites without leaving the application. You will make the Web Browser form a complete browsing tool that allows the user to browse to other Internet Addresses.

➢ **Create the browser form**

1. If it isn't already open, open the Visual Basic project, Leads.vbp.
2. Add a new Form to the project, named frmBrowse.
3. Place controls on the form so that it looks like the diagram below. Select a control in the diagram to see the type of control and the name used in the solution files.



4. Create the Click event procedure for the Go command button:
   a. Delete the **Private** keyword from the cmdGo_Click Sub procedure.

Note In the next step, you will call this event procedure from the frmLeads form. In order for you to call this event procedure from a different form, it cannot be **Private**.

---

   b. Read the address from the Address text box.
   c. Invoke the **Navigate** method of the WebBrowser control, passing that address.
5. Save the new form.

➢ **Implement the Open Web Site in VB menu item**

1. Open the code window for the frmLeads form and locate the OpenWebInVB Sub procedure.

   This procedure is called from both of the Open Web Site in VB menu items.

2. Read the Web site address from the selected customer in the lstLeads list view control. The Web site address is SubItem 5:

   ```
   lstLeads.SelectedItem.SubItems(5)
   ```

3. Set the Address text box, on the frmBrowse form, to the Web site address.
4. Invoke the Click event of the Go command button on the frmBrowse form.
5. Save your changes.

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

```
Private Sub OpenWebInVB()
    frmBrowse.txtAddress.Text = lstLeads.SelectedItem.SubItems(5)
    frmBrowse.Show
    frmBrowse.cmdGo_Click
End Sub
```

➢ **Test the Open Web Site in VB menu item**

1. Run the application.
2. Select Read Leads List from the File menu.
3. Select a customer in the list.
4. Select Open Web in VB from the File menu.
5. On the frmBrowse form, enter a different Web site address in the text box and choose Go.

➢ **Make the browser form more complete**

1. Add event procedures for the Click event of the Forward and Back command buttons.

   Before you can test either of these buttons, you need to go to a second Web site.

2. Add an event procedure for the Click event of the Refresh command button.
3. When the status text changes in the WebBrowser control (the OnStatusTextChange event), reflect that change in the browser form's status bar.
4. Keep the Address text box current with the address of the page displayed, by setting its Text property in the OnNavigate event of the WebBrowser control.
5. Save your changes, and test the new functionality.