

Building a Browser

D's Browser is a fully functional Internet browser I created. You ask, "What is a browser?" A browser is a program that can interpret hypertext code used to render web pages, pictures and other multimedia. I started making D's Browser in February 2002 after I got the idea from a training video my friend and I were watching. I decided to make myself a browser with links to the web sites that I visited the most.

Using Visual Basic 6, I repeated the procedure that I saw in the movie. This was not enough to make the browser a finished program without errors; therefore, I had to do more. I was not sure what I could do, and I was unaware of how to do the things I wanted to implement in my project. My lack of knowledge made production slow but my determination made me keep adding to my Browser. After some time, I decided to show my teacher, Mr. Barhoum, my personal project for some help and he advised me to use it for my graduation project. I was excited to hear I could use something that I was making for fun, as school work. I spent many days and late nights working on my creation. After I learned new things in class, I added more functionality to the program. For this report I broke down the steps of this project as follows.

1. Load Components into Toolbox
2. Design Interface & Write Code
3. Test and Fix Errors
4. Make executable file

Now it is time to get started creating a browser of your own. I have included screenshots that are very helpful for beginning programmers.

1. Load Components into Toolbox

Open Visual Basic and highlight the "Standard EXE" as shown in Figure 1.0 and click "Open" to create a new standard EXE. If this dialog does not appear select "New Project" from the "File" menu



Figure 1.0



Figure 1.1

There are three control components that are needed for this project, “*Microsoft Windows Common Controls 6.0*” “*Microsoft Common Dialog Control 6.0*” and “*Microsoft Internet Controls*”. The controls are not all visible at the same time so I have created Figure 1.2a through Figure 1.2c that show what components are needed. Select these three items, click “Apply” then click the “Close” button.

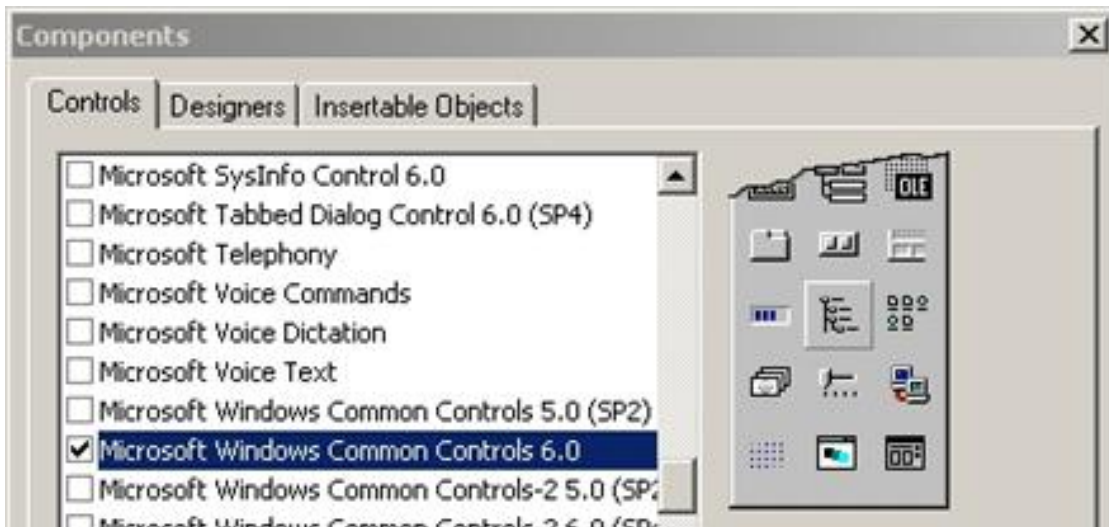


Figure 1.2a



Figure 1.2b



Figure 1.2c

Notice the change in the toolbox. There are now more buttons than before. You can put the mouse over the new buttons to find out the name of each of them. Here are a few of the buttons that are added into the toolbox. In Figure 1.5a the button on the left is the Toolbar Control and the one on the right is the StatusBar Control. Figure 1.5b is the WebBrowser Control.



Figure 1.3a



Figure 1.3b

2. Design Interface & Write Code

The next step is to design and code the interface for your browser. Start by adding a toolbar. Double click on the “*toolbar*” button in the toolbox and the toolbar will be placed on the form for you. The other object-controls to add are: *ComboBox*, *ImageList*, *WebBrowser*, and a *StatusBar*. All of these objects can be added by double clicking on their button. Using naming conventions, give these objects names such as *cboURL* for the *address bar* and *tlbBrowser* for the browsers *toolbar*. The *ImageList* should be named *imgList*. After all Object Controls have been named and arranged accordingly (*basically however you want*) it is time to write the code.

You should have something like Figure 2.0.

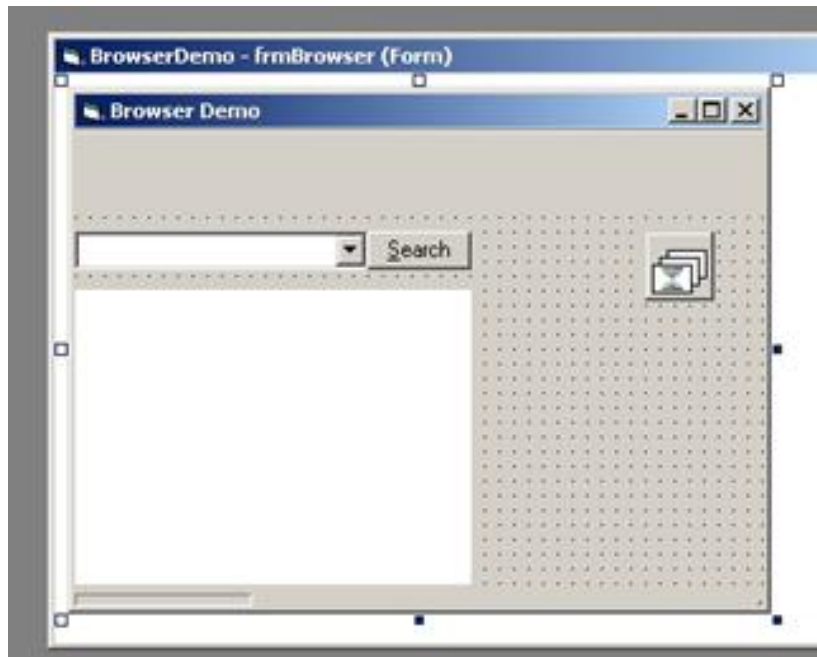


Figure 2.0

Before you write the code for the controls you placed on the form, load pictures into the *ImageList*.



Select the *ImageList* and double click on “*custom*” in the properties menu. If the properties window is not on the screen press the **F4** key and it will appear on the screen. The second tab is for images, go there and click “Insert Picture”. Each picture must have a “*Key*” and a “*Tag*”. I have used the same name for both the Key and the Tag. *For example the back arrow has the Key of Back and the*

Tag of Back. The Home arrow has a Key of Home and a Tag of home as shown in Figure 2.1.

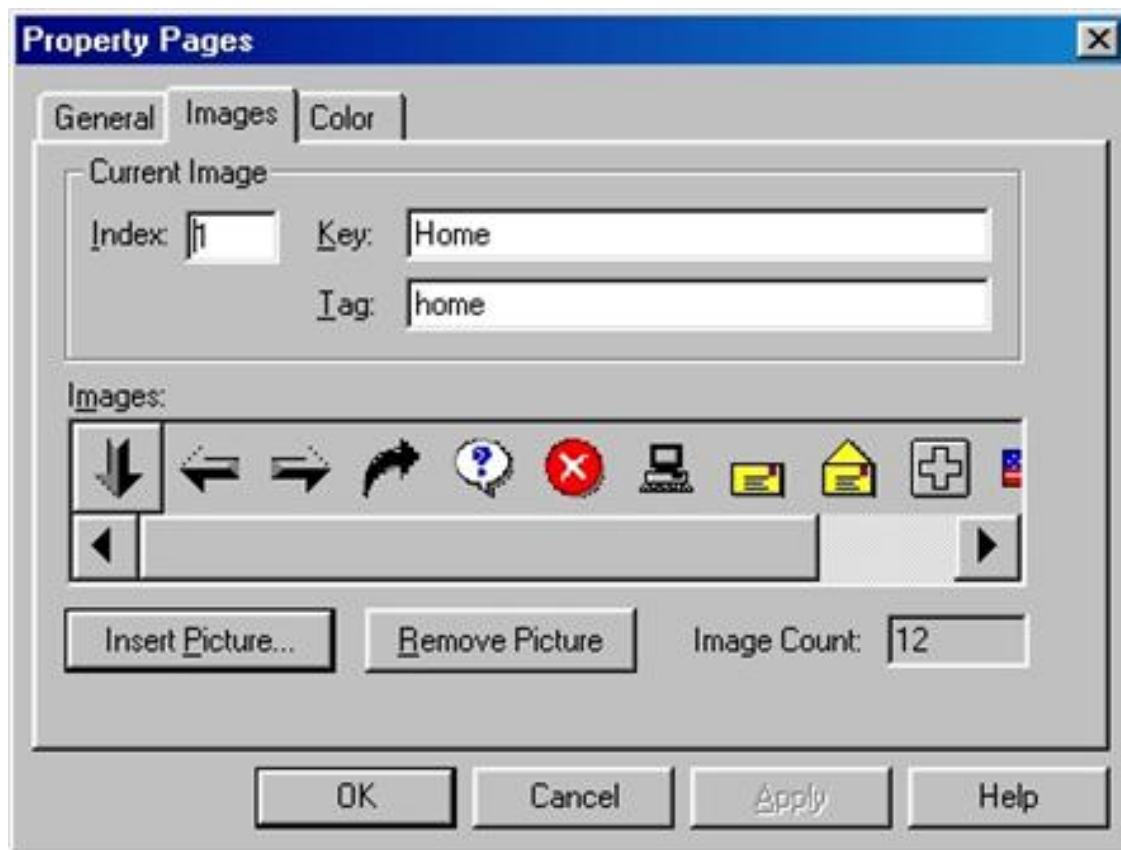
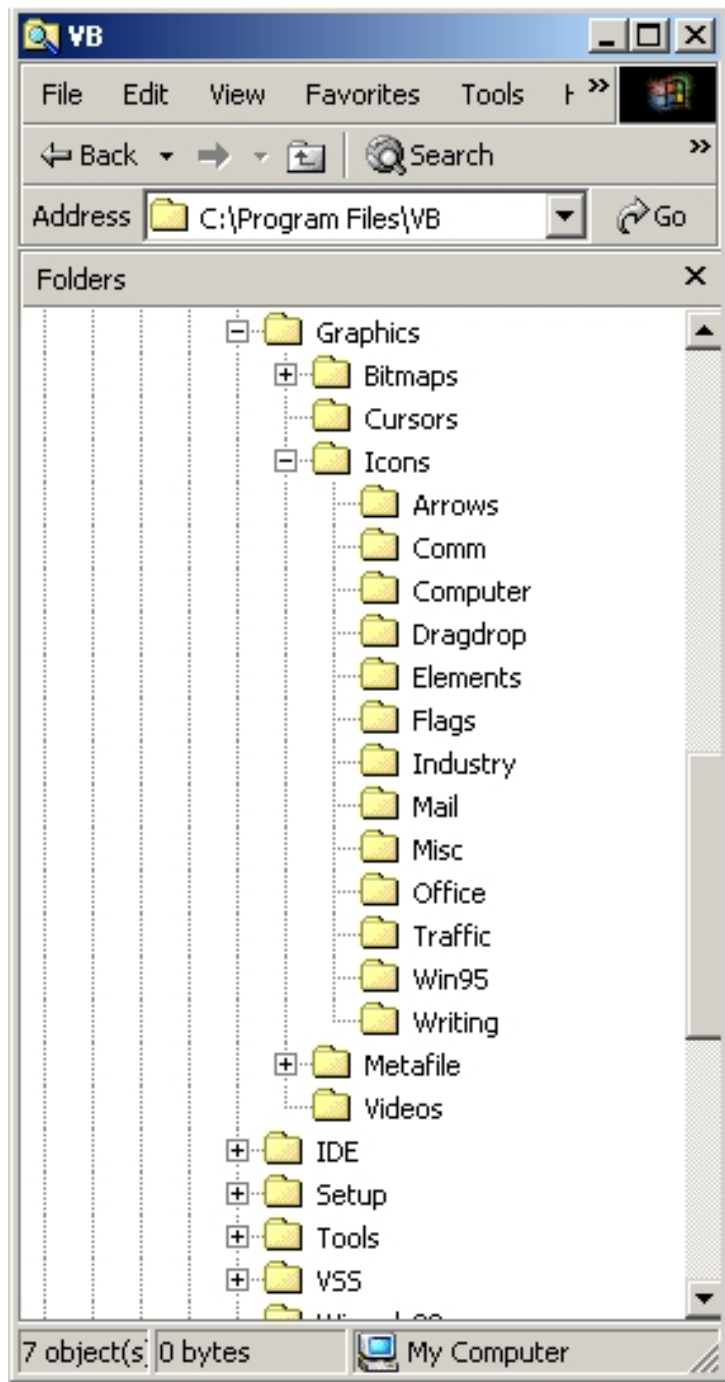


Figure 2.1

There are some generic icons that come with Visual Basic. They are located in the programs folder in a folder called graphics as shown in Figure 2.2. I don't think these pictures are very helpful for buttons to web sites; however, they can be good for navigational controls.

**Figure 2.2**

The Image List must be linked to the toolbar before you can use the pictures that you loaded into it. Select the toolbar on your browser form and double click “custom” in the properties window. In the General Properties Tab, the dropdown box is circled where you need to bind the Image List. Click on

the dropdown button and select the Image List. I have already done this in the picture shown Figure 2.3. The other options that are underlined are for the appearance of the toolbar. You can play with these options to see which way you like best.

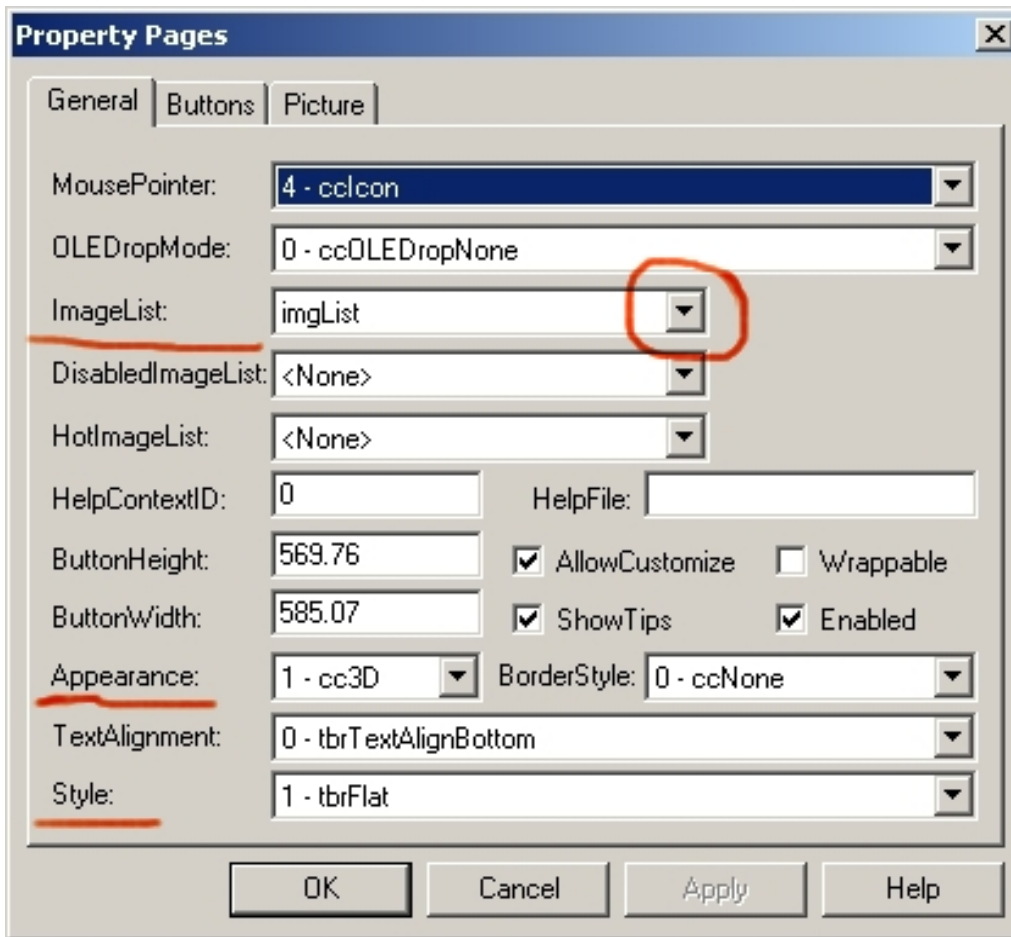


Figure 2.3

If you added a status bar, it should be at the bottom of the form by default. Click on it and then double click on “custom” in the properties menu. Click on the “Panels” tab then click “Insert Panel” until there are about four of them. The style can be changed to several default types including caps lock, date, and time by changing the value in the “Style” dropdown box. Configure the other buttons by

changing the index with the arrows. Pictures and text can also be added. Refer to Figure 2.4 for location of these properties.



Figure 2.4

Now it is time to write the code. Since this project is going to be a browser, let's start with displaying a web page in the browser control. Double click on the search button and enter this code in the click event.

wbBrowser.Navigate cboURL.Text

With this code the assumption is that the name of your browser control is wbBrowser and the combo box for the address is named cboURL. The .Navigate method makes a browser go to a defined location. The location is defined by the text in the address bar "cboURL." You can test to see if it works; press **F5** on the top row of the keyboard to run the program. Type your favorite URL in the address box and click on the search button with your mouse. Did it work? If not, your naming might not be the same as mine; check the names of your objects. Press **F7** if the code window is not already showing and type the following code.

Private Sub Form_Load ()

''Adjust objects according to size of form

```
tlbBrowser.Refresh  
Form_Resize
```

```
''Use my homepage  
cboURL.Text = "http://www.dowtech.us/"  
wbBrowser.Navigate cboURL.Text
```

End Sub

Private Sub Form_Resize ()

```
''Error correction and resize form & objects  
On Error Resume Next
```

```
wbBrowser.Width = Me.ScaleWidth  
If frmBrowser.ScaleHeight > 0 Then  
    wbBrowser.Height = (Me.ScaleHeight - tlbBrowser.Height) - 600  
End If
```

End Sub

Private Sub tlbBrowser_ButtonClick(ByVal Button As MSComctlLib.Button)

```
''Toolbar buttons in the order as they appear  
On Error Resume Next
```

```
Select Case Button.Key
```

```
Case "back"  
    wbBrowser.GoBack
```

```
Case "forward"  
    wbBrowser.GoForward
```

```
Case "fresh"  
    wbBrowser.Refresh
```

```
Case "home"  
    cboURL.Text = "http://www.dowtech.us/"  
    wbBrowser.Navigate cboURL.Text
```

```
''to use Internet Explorers home page use this  
''code instead of the two lines after the Case "home"  
''''''wbbrowser.gohome
```

```
Case "stop"
  wbBrowser.Stop
```

'''Example for web link buttons

'''Change "favoriteURL" to your favorite page name

```
Case "favoriteURL"
```

```
  cboURL.Text = "http://www.favoriteURL.com/"
```

```
  cmdSearch_Click
```

End Select

End Sub

Once you have that code down and the naming is the same between the objects and the code press **F5** and your browser should look like Figure 2.5. (*Scrollbar is blue because the web site is designed that way.*) **Note:** Icons may not be the same.

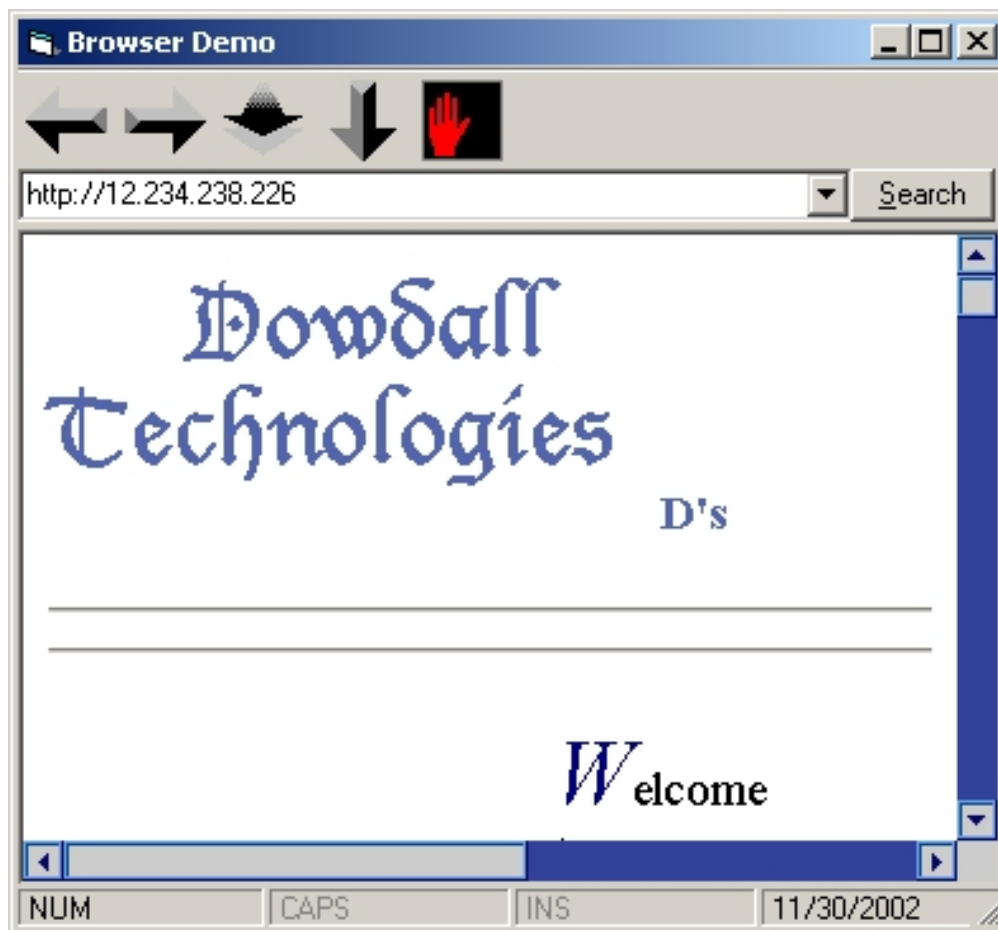


Figure 2.5

As you can see in Figure 2.5 the major foundation of the browser is done. More buttons can be added to the toolbar after icons are picked out. To have buttons linked to web addresses use this code.

Case "google"

```
cboURL.Text = "http://www.google.com/"
cmdSearch_Click
```

For the program to remember what web pages it was at after its closed, data must be saved to a file. The file must then be opened when the program starts again. (*For this code to work you must have a file named "url.dat" in the folder the program is contained in.*) To make this file we will run the program once after we add the "save" code. This will make the file for us. Then we will add the code to "open" the file. To get started double click on the form to open the "Form" event code section. Go to the Form Query Unload event by changing the value in the dropdown box at the top right of the code window. Refer to Figure 2.6



Figure 2.6

Enter the following "save" code in that event

```
''Write address list to data file
Dim intURLIndex As Integer
  Open App.Path & "\url.dat" For Output As #12
  For intURLIndex = 0 To cboURL.ListCount - 1
    Write #12, cboURL.List(intURLIndex),
  Next
Close #12
```

Press **F5** to run the program once then close it. Open the folder that contains your program and look for the file named "url.dat" If it's there then enter the following code before the "End Sub" in the "Form Load" event then run the program again.

```
''load previously entered URL's
Dim strURL As String

  Open App.Path & "\url.dat" For Input As #10
  Do Until EOF (10)
    Input #10, strURL
    cboURL.AddItem strURL
  Loop
Close #10
```

Run your program again by pressing **F5** to test if it works. If it does not, make sure the “url.dat” file is in the folder your program is saved in.

One thing that gave me a problem was trying to stop the address box from having multiple identical entries. At first I did not know where the code was supposed to go. Then when I entered it in the click event of the “Search” button I had errors with the way it was written. I was very happy when this was fixed; I had spent a lot of time working on it.

Now that I’ve done all of the figuring it’s time for you to start adding the code. Double click on the “search” button. By default the “click” event is opened in the “Code Window” modify the code to duplicate the following:

Private Sub cmdSearch_Click ()

```

'Go to location
    wbBrowser.Navigate cboURL.Text

'Locate the first matching URL occurrence in the list or add address
    Dim blnItemFound As Boolean
    Dim intItemIndex As Integer
    blnItemFound = False
    intItemIndex = 0
'If address box is not blank then look for duplicate
    If cboURL.Text <> "" Then
        Do Until blnItemFound Or intItemIndex = cboURL.ListCount
If Trim (UCase (cboURL.Text)) = Trim (UCase(cboURL.List(intItemIndex))) Then
            blnItemFound = True
            End If
            intItemIndex = intItemIndex + 1
        Loop
        If blnItemFound = False Then
            With cboURL

                .AddItem .Text
                wbBrowser.Navigate .Text
            End With
        End If
    End If
End Sub

```

With this code, the assumptions are that the browser control is named “wbBrowser”, the address box is

named “cboURL”. This code searches the list property of the address bar for a duplicate value. If there are no identical entries, the value is added to the list. Go visit some web sites in your browser to test the addition you just made. After you typed in a couple different web addresses into the address bar, retype them the same way. Now click the dropdown list in the address box to see if there are multiple duplicate entries.

3. Test and Fix Errors

One of the hardest things for new programmers to do is fix errors. Errors can be really hard to find if code does not make sense to you. Names can be spelled wrong or have extra characters in them from typing errors. The easiest way to find an error is to run the program. A lot of errors will be found this way but some will not be found until after actually using the application for a while. Testing any program that you made is a very important thing to do before trying to distribute it. If people use a really bad version of your software, they will think that your company does not know what it is doing. They will not want to come back to you for their software needs. Testing must be repeated after fixing errors until everything works correctly.

4. Make Executable File (EXE)

Now you’re done writing all the code for the interface you designed. You tested and fixed all known errors. It is time to compile your program into an executable file. Click on the “File” menu and select make *.EXE from the list. Where the * will be the

name of your project. The .EXE file will have to be packaged with the .dat files that are part of your program. For example: “url.dat” must be included in the package. The package I am referring to is a compressed folder containing all the files necessary to make the program work. To do this packaging you must have a compression utility installed on your PC. I use WinZip for my compression needs. If you are using WinZip, right click on the folder your files are in and select “Add to *.Zip”. Where * would be the name of your project.

Conclusion

During the past ten months of designing D’s Browser, I have learned many things about Object-

Oriented-Programming. One of the things I learned how to do was access sample code from the internet for use in my project. The code must be reviewed and modified to work with existing projects. *For example, the form names can not be the same when joining two projects together.* Most important of all I learned how to implement all my skills into one project that has multiple functions. D's Browser has been the most exciting creation I have ever made.

Bibliography

Books:

1. Bradley, Julia C. and Millspaugh, Anita C. Programming in Visual Basic 6.0. New York: McGraw-Hill Irwin, 2002
2. Deitel, H. M. How to program Visual Basic.Net New Jersey: Prentice Hall, 2002
3. Shammas, Namir. Visual Basic 6 for Dummies California: IDG Books Worldwide, Inc 1998

Internet Sources:

1. "Microsoft Developer Network" 2002. Feb-Dec 2002.

<<http://msdn.microsoft.com/>>

2. Richmond, Allen. “Web browsers” 2002. Oct 2002.

<<http://www.wdvl.com/Software/Browsers/>>

3. “MSDN” How to Print content of web browser control. 2001. Feb-Dec 2002.

<<http://support.microsoft.com/default.aspx?scid=KB;en-us;q188271>>

4. “CodeArchive” HTMLViewerSample. 1999. Nov 2002.

<<http://www.codearchive.com/vbasic/internet.html> >

5. “MSDN” Navigate Complete event. 2002. Feb-Dec 2002.

<<http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/webbrowser/reference/events/navigatecomplete2.asp>>

6. ”MSDN” Go Back Method. 2002. Sep 2002

<<http://msdn.microsoft.com/workshop/browser/webbrowser/reference/methods/goback.asp>>

Glossary

Browser - A browser is a program that can interpret hypertext code used to render web pages, pictures and other multimedia.

Component – An element of a software system that increases functionality.

.EXE – A file that is executable on Windows operating systems.

Hypertext – A computer based text code system used to display pictures and words.

Interface – A shared boundary between the user and the software.

E.g. - Graphic Interface

Multi-media – The combined use of different types of media in which communication is carried.

URL – The name based address of an Internet location such as a web site.
Uniform Resource Locator.