

Integrating HTML Help into your applications

by Sant S. Pramoda

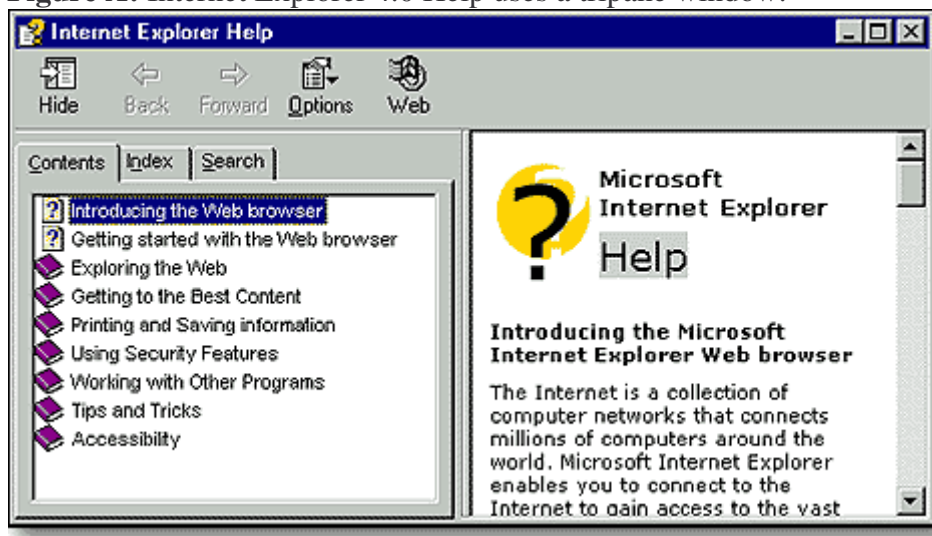
Have you ever peeked into the Internet Explorer (IE) 4.0 Help and wondered how to build a Help system like that for your application? In this article, we'll show you how to build an HTML-based help system using Microsoft's HTML Help Workshop. We'll also explore how to call the Help file you built from your VB application by using HTML Help Workshop--you'll build a simple `CHelp` wrapper class to access the Help files.

Microsoft introduced HTML Help in August 1997 as a successor to WinHelp. Since it's relatively new, it hasn't been used in many applications other than IE. However, it's expected that all Microsoft's future products will use HTML Help, starting with Windows 98. You need to have IE 3.02 or higher installed in order for HTML Help to work; to take advantage of the full functionality, you need the latest version of IE.

The basics

A typical HTML Help window consists of three panes, as shown in Figure A. (This is sometimes referred to as a tripane window.) the top of the window is the ToolBar pane, which contains the toolbar buttons. To the bottom-left is the Navigation pane; it's made from the `HHCtrl.OCX` ActiveX control that's responsible for Contents, Index, and Search capabilities. The bottom-right pane is the Topic pane, which acts as a container for HTML documents.

Figure A: Internet Explorer 4.0 Help uses a tripane window.



HTML Help Workshop

To author HTML Help files, you can use HTML Help Workshop, which is a tool developed by Microsoft. HTML Help Workshop is available for download from www.microsoft.com/workshop/author/htmlhelp/default.asp. In this section, we'll show you how to build a simple Help system using this tool. Before you begin, you need to install HTML Help Workshop on your machine. For a description of the different file types and extensions used in the HTML Help, please refer to Table A.

Table A: Files used in HTML Help

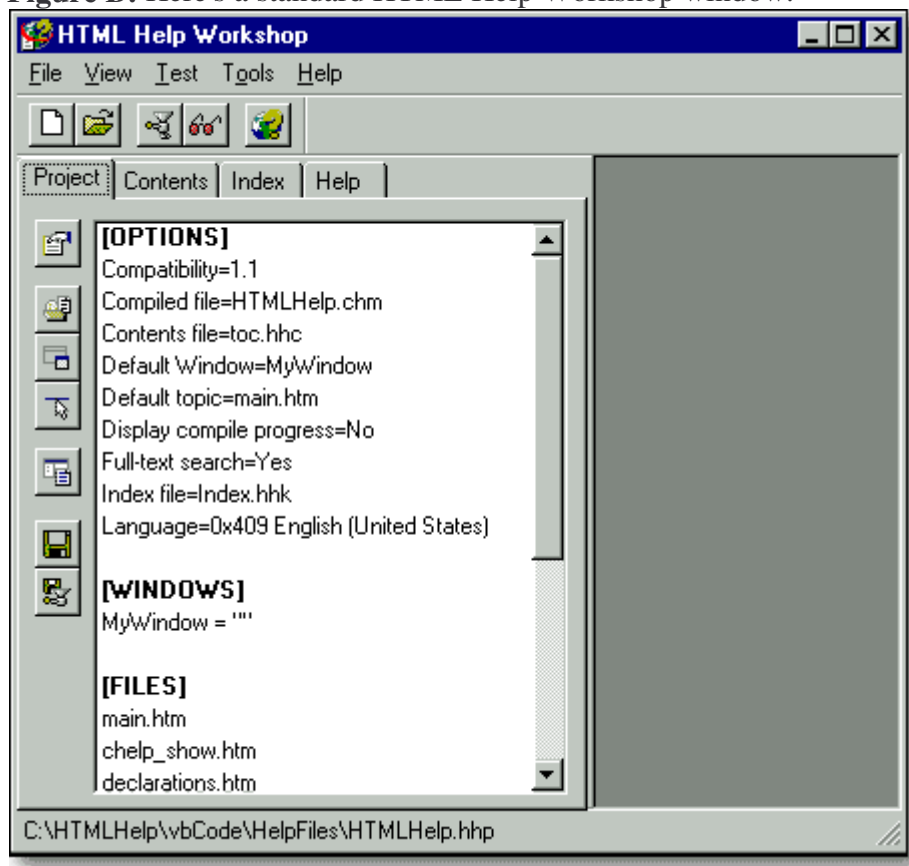
| Extension | Type |
|-----------|-------------------|
| HHP | Help project |
| HHC | Table of contents |
| HHK | Index |
| CHM | Compiled Help |
| HTM | Individual Help |
| H | Header files |

Creating your first HTML Help project

Before you can build your Help system, you'll need to create all the HTML files that the system will need. Then, create an \HTMLHelp directory and save the HTML files there--this will be your project directory.

Now, open HTML Help Workshop and choose the File | New... menu item. In the resulting New dialog box, select Project and click OK. The New Project Wizard will start. Follow the steps in this Wizard to create your project. Be sure to check the HTML Files option in the New Project-Existing Files dialog box, to add your HTML files to the project. Click the Finish button to create your project. Figure B shows a typical HTML Help Workshop window.

Figure B: Here's a standard HTML Help Workshop window.



The HTML Help Workshop's main window consists of four tabs--Project, Contents, Index, and Help--and a set of buttons for each tab. You can use the Project tab to change project options, add or remove topic files, add or modify window definitions, and set options for the `HTMLHelp` API. Setting up Window definitions and `HTMLHelp` API are important for VB developers. We'll explore how to set these options later.

Adding table of contents and index files

A table of contents file defines the contents you see when you click the Contents tab in a Help file. The contents are usually organized in a tree structure, with headings, subheadings, and pages.

To add a table of contents file to your project, click the Contents tab to see the Table Of Contents Not Specified dialog box. Select Create A New Contents File to create a blank table of contents file. Now, you can add your contents using the Insert Heading and Insert Page buttons. In the dialog boxes that appear when you select these options, you can associate HTML files to the headings and pages. You can also edit, delete, or organize your headings and pages using the Edit, Delete, and Move Selection buttons.

The index file defines the index entries (keywords) you see when you click on the Index tab. To add an index file to your project, click on the Index tab to open the Index Not Specified dialog box. Select Create New Index to create a blank index file. You can add your keywords using the Insert A Keyword button and associate an HTML file for that keyword. You can also edit, delete, or organize your keywords using the Edit, Delete, and Move Selection buttons.

Window definitions and HTMLHelp API information

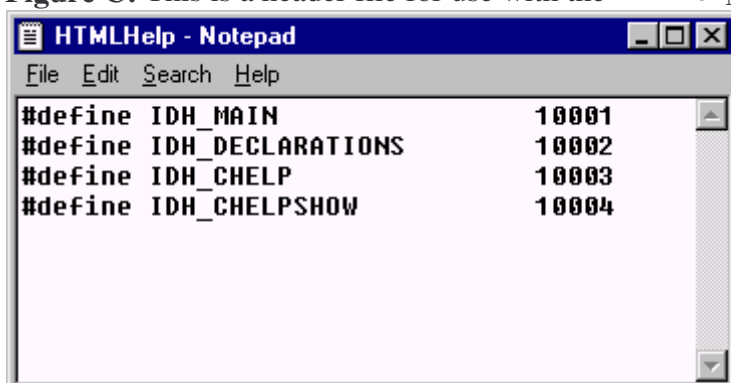
Window definitions let you define the appearance of a help window. You can specify which buttons to show on the toolbar, which tabs to show in the navigation pane, and a set of other properties, which alter the look of a Help window. You can set up as many window definitions as you want, but you must have a default window definition.

When you call the HTMLHelp API from VB, you can refer to these window definitions and show the Help window in different styles. Click on the Project tab, then click the Add/Modify Window Definitions button. In the Add A New Window Type dialog box, type *MyWindow* (or any other name) and click OK. In the Window Types dialog box, you can select different tabs to change the buttons, styles, and so on. When you click OK, a new window definition will be created.

If you need to include a Search tab in your Help file, click the Navigation Pane tab and select the Add Search Tab option. When you click OK, you'll be taken through a Wizard. In the second step, select Compile Full-text Search Information to let the Help compiler build the entries for the Search tab when you compile the project.

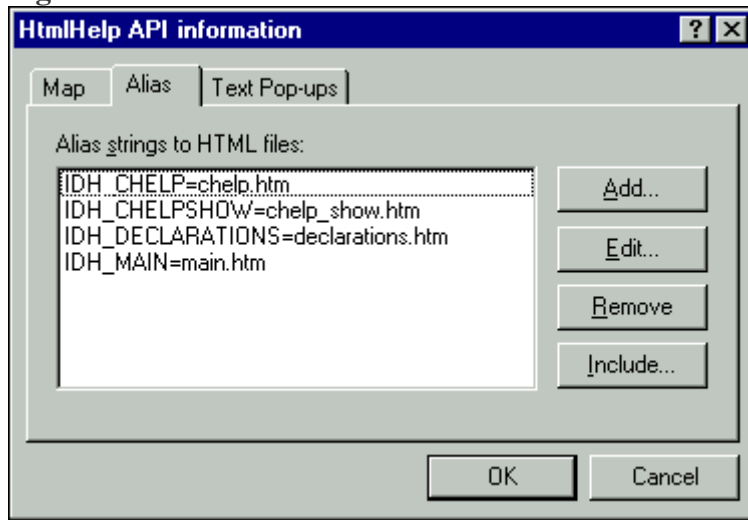
The HTMLHelp API Information button sets up information for context-sensitive Help. When you click this button, you'll see the HTMLHelp API information dialog box. You can associate a header file--which defines all context IDs--in the Map tab. The current version of HTMLHelp accepts only C-type header files. Figure C shows the contents of a typical header file in Notepad.

Figure C: This is a header file for use with the HTMLHelp API.



You can use the Alias tab to associate the Help context ID constants defined in the header file to HTML files, as shown in Figure D. When you call `HTMLHelp` from VB with a Help context ID constant, you can navigate to the HTML file associated with that constant.

Figure D: Associate context IDs with HTML files in the Alias tab.



You can also set up text pop-ups using the Text Pop-ups tab. For more details on setting up text pop-ups, refer to the online Help that comes with the HTML Help Workshop.

Compiling your Help project

Before you compile, click the Change Project Options button in the Project tab and select the Default File for your project in the General tab. You can also click on the Files tab to change the name of the compiled file or to verify the table of contents and index files. Click the Save All Project Files And Compile button to compile your file.

By default, your compiled help file will have the same name as your project file--but it will have a CHM extension. If there are any errors during compile, you'll see them in the window on the right side. Those errors are self-explanatory. After a successful compilation, you're ready to call your Help system from VB.

Calling Help from VB

You can use the `HTMLHelp` API function to display Help files from VB. `HTMLHelp` is part of the `hhctrl.ocx`. It's modeled after the `WinHelp` API function--so, if you're familiar with `WinHelp`, you won't have any difficulty transitioning to `HTMLHelp`. In this section, we'll show you how to wrap the `HTMLHelp` API in a class (`CHelp`) so that you can reuse it.

To begin, create a class and name it `CHelp`. Enter the following code in the General Declarations:

```
// Constant declarations
Private Const HH_DISPLAY_TOPIC = &H0
Private Const HH_HELP_CONTEXT = &HF

// API declaration
Private Declare Function HtmlHelp Lib _
    "hhctrl.ocx" Alias "HtmlHelpA" _
    (ByVal hwndCaller As Long, _
    ByVal pszFile As String, _
    ByVal uCommand As Long, ByVal dwData As Long) _
    As Long
```

The `CHelp` class has one method--`Show`, in Listing A--that takes a filename as its first argument. This filename could be a compiled file (**CHM**) or an HTML file (**HTM**). In case of a compiled file, you can use the next two arguments to further manipulate the Help window. The second argument, `WindowPane`, is optional. It refers to the window definitions we discussed earlier, and can be used to show different window types. The third argument is the `ContextID`, which is also optional. You can use this argument to supply `ContextID` when context-sensitive help is called.

Listing A: `CHelp_Show` method

```
Public Sub Show(NewFile As String, _
    Optional WindowPane As String, Optional ContextID)

Dim strFile As String
Dim hRet As Long

strFile = NewFile

If Len(WindowPane) Then
'you need to include > symbol with the file name
    strFile = Trim(strFile) & _ ">" & Trim(WindowPane)
End If

If IsMissing(ContextID) Then
    hRet = HtmlHelp(0, strFile, _ HH_DISPLAY_TOPIC, _
        ByVal 0&)
Else
    hRet = HtmlHelp(0, strFile, _ HH_HELP_CONTEXT, _
        ContextID)
End If
End Sub
```

As you can see in the listing, you must include a `>` symbol with the filename, if you're using a `WindowPane`. The `HH_DISPLAY_TOPIC` command displays a topic in the Help window.

`HH_HELP_CONTEXT` displays topics associated with a `ContextID`. On success, the `HTMLHelp` API will return the handle of the Help window created.

Using the `CHelp` class

Once you create an instance of the `CHelp` class, all you need to do is call the `Show` method and pass the name of the compiled file as the parameter. You can also pass the name of an HTML file to open it directly in the Topics pane without the Navigation and ToolBar panes.

Listing B shows how to call the `Show` method in the `Click` procedure of a button. You can play with other parameters to manipulate the Help window.

Listing B: Using `CHelp` class

```
Private Sub cmdShow_Click()

Dim strFile As String
Dim objHelp As CHelp

Set objHelp = New CHelp

strFile = App.Path & "\HelpFiles\HTMLHelp.chm"

Call objHelp.Show(strFile, "MyWindow")

Set objHelp = Nothing
End Sub
```

Conclusion

HTML Help is definitely the Help of the future--it takes a normal Help system to a new level. Since it's HTML based, you can mix and match pictures, movies, animation, and multimedia to create a rich set of Help files. The possibilities are endless.