

The HTML Help class was developed in Microsoft Visual Basic. It implements the major functions of HTML Help for Visual Basic applications, and has been tested under both VB5 and VB6.

In order to use the HTML Help class for VB, use Add Class Module from the Project menu in Visual Basic to add the clsHTMLHelp.cls to the project. Once the class is part of the project, create a new object of the type HTMLHelp.

In order to use the HTML Help Object Library with VB or VBA, install the DLL and related files onto the development computer. Within Visual Basic or a VBA application, select References from the Project directory, and click the box next to HTML Help Object Library. It may be necessary to use the Browse button in the References dialog to find the help.dll file.

The declaration only needs to be done once when the application starts, but in an effort to save on memory usage, it's suggested that it only be done when necessary. It needs to be created either as a module-level object, as in:

```
Private hHelp As New HTMLHelp
```

or as a global object, like this:

```
Public hHelp As New HTMLHelp
```

Examples throughout this documentation will show creation of a module-level object.

Calling HTML Help Files

In calling a Help file via the HTML Help class, set the properties of the object according to what you want to accomplish. Calling a particular topic from a **Help** button on a form would look like this:

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicID = 1030
        .HHWindow = "Main"
        .HHDisplayTopicID
    End With

End Sub
```

Calling HTML Help from a **Help...Contents** menu item, and forcing the Contents tab to be displayed in a predefined tri-pane window, use code similar to:

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpContents_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplayContents
    End With

End Sub
```

```
End Sub
```

Similar code can be used to force the Index tab to be displayed:

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpIndex_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplayIndex
    End With

End Sub
```

Once the form closes, or when the application closes, make sure to clean up all open HTML Help windows and set the object to Nothing:

```
Private hHelp As New HTMLHelp
Private Sub Form_QueryUnload _
(Cancel As Integer, UnloadMode As Integer)

    hHelp.HHClose
    Set hHelp = Nothing

End Sub
```

Checks the registration of the specified CHM file. If the file is found to be registered, the `HHRegFileName` and `HHRegFilePath` properties are loaded with the found information. Once the information is collected from the registry, the file is located. If it's in the registered location, `HHRegFileExists` is set to `True`. Otherwise, `HHRegFileExists` is set to `False`. If the file is not registered, the `HHRegFileName` and `HHRegFilePath` properties return empty strings.

Syntax

object.**HHCheckRegistry** (*file* as string)

object An object expression that evaluates to an instance of the `HTMLHelp` object.

file The file to find in the registry.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileCheckRegistry_Click()
    Dim strFileToCheck As String
    strFileToCheck = txtFileName.Text
    hHelp.HHCheckRegistry strFileToCheck
    If hHelp.HHRegFileName = "" Then
        MsgBox strFileToCheck & " is not registered.", _
            vbExclamation + vbOKOnly, _
            "Registry Check Result"
    Else
        MsgBox strFileToCheck & " is registered as " & _
            & hHelp.HHRegFilePath & Chr(10) & _
            "File exists: " & hHelp.HHRegFileExists, _
            vbInformation + vbOKOnly, _
            "Registry Check Result"
    End If
End Sub
```

Remarks

When an HTML Help file is listed in the registry, it's registered as:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\HTML Help]
```

```
Name = myfile.chm
```

```
Value = "c:\chmfolder"
```

This allows for opening the CHM without specifying the path, as in:

```
hhhelp.CHMFile = "myfile.chm"
```

and

```
c:\windows\hh.exe myfile.chm
```

On Windows NT, this is generally seen as:

```
c:\winnt\hh.exe myfile.chm
```

Closes all open HTML Help windows.

Syntax

object.**HHC**lose

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub Form_QueryUnload_
(Cancel As Integer, UnloadMode As Integer)

    hHelp.HHClose

End Sub
```

Runs HH.exe and displays the HTML Help topic specified in the **HHALink** property.

Syntax

object.**HHDisplayALink**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp

Private Sub cmdALink_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicURL.Text
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = txtTitle.Text
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHKeyword = txtALink.Text
        .HHDisplayALink
    End With

End Sub
```

Runs HH.exe and displays the HTML Help file you specify, forcing the file to display the Contents tab.

Syntax

object.**HHDisplayContents**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpContents_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplayContents
    End With

End Sub
```

Runs HH.exe and displays the HTML Help file you specify, forcing the file to display the Index tab.

Syntax

object.**HHDisplayIndex**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpContents_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplayIndex
    End With

End Sub
```

Runs HH.exe and displays the HTML Help topic specified in the **HHKeyword** property.

Syntax

object.**HHDisplayKeyword**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp

Private Sub cmdKeyword_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicURL.Text
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = txtTitle.Text
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHKeyWord = txtKeyword.Text
        .HHDisplayKeyword
    End With

End Sub
```

Displays a text popup at the current coordinates of the mouse pointer. The text can come from a string, a resource file, or a file listed in the [TEXT POPUPS] section of a CHM's HHP file.

Syntax

object.**HHDisplayPopup**(*form hWnd*)

object An object expression that evaluates to an instance of the HTMLHelp object.

form hWnd The handle of the calling form.

Examples

The first example illustrates how to create a simple popup from a string containing the desired information. Note the use of Chr(10) to create multiple paragraphs in the popup:

```
Private Sub mnuExamplesTextPopup_Click()  
    Dim strPopup As String  
    With hHelp  
        .HHPopupType = HH_TEXT_POPUP  
        .HHPopupCustomColors = False  
        strPopup = "This is a test using a simple text string." & _  
            Chr(10) & Chr(10) & _  
            "Notice how you can have more than one paragraph."  
        .HHPopupText = strPopup  
        .HHDisplayPopup Me.hWnd  
    End With  
End Sub
```

The next example demonstrates getting the text for the popup from a resource file. In this case, the resource ID is 101. This also demonstrates the use of custom colors for the popup text and background. These colors are in the form &HBBGRR.

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFFF0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

The final example illustrates providing the popup for the text from the popup topics in a CHM file. Also shown are the use of standard colors for the popup's text and background, and a custom font.

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)
```

```
.HHPopupID = 21000
.HHPopupCustomColors = False
.HHPopupBackColor = vbBlack
.HHPopupTextColor = vbWhite
.HHPopupTextFont = "Comic Sans MS"
.HHPopupTextSize = "8"
.HHPopupTextBold = True
.HHPopupTextItalic = True
.HHPopupTextUnderline = True
.HHDisplayPopup Me.hwnd

End With

End Sub
```

In this situation, HHSetHelpFile is modified to also contain the name of the popup file compiled into the CHM:

```
Public Function HHSetHelpFile(ByVal intSelHelpFile As Integer) As String
    ' Set the string variable to
    ' include the application path

    Select Case intSelHelpFile
    Case 1
        HHSetHelpFile = App.Path & "\Library.chm"
    Case 2
        ' Popup text file for the above CHM
        HHSetHelpFile = "Library.txt"
    Case Else
        ' list other HTML Help files here
    End Select

End Function
```

Runs HH.exe and displays the HTML Help file you specify, forcing the file to display the Search tab.

Syntax

object.HHDisplaySearch

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpSearch_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplaySearch
    End With

End Sub
```

Runs HH.exe and displays the HTML Help topic specified in the **HHTopicID** property.

Syntax

object.**HHDisplayTopicID**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicID = 1030
        .HHWindow = "Main"
        .HHDisplayTopicID
    End With

End Sub
```

Runs HH.exe and displays the HTML Help topic specified in the **HHTopicURL** property.

Syntax

object.**HHDisplayTopicURL**

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicURL = "html\introduction.htm"
        .HHWindow = "Main"
        .HHDisplayTopicURL
    End With

End Sub
```

Converts the mouse pointer to the What's This mouse pointer. When the left button is then clicked on a control, What's This is invoked for that control.

Note: This method does not exist in the HTML Help Object Library DLL.

Syntax

object.**HHInvokeWhatsThisHelp** (*hWnd*)

object An object expression that evaluates to an instance of the HTMLHelp object.

hWnd The handle of the calling form.

Example

This method is particularly useful for invoking What's This Help from a menu item, generally Help ... What's This?

```
Private Sub mnuHelpWhatsThis_Click()  
    hHelp.HHInvokeWhatsThisHelp Me.hwnd  
End Sub
```

Registers the specified CHM file. This method calls the HHCheckRegistry method at the end of its routine as verification of the registration process. If the specified file is already registered before HHRegister is called, an error message is generated.

Syntax

object.**HHRegister** (*file as string*)

object An object expression that evaluates to an instance of the HTMLHelp object.

file The CHM file, including the path, to register in the registry.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileRegister_Click()
    Dim strFileToRegister As String
    strFileToRegister = txtFileName.Text
    hHelp.HHRegister strFileToRegister
End Sub
```

Remarks

When an HTML Help file is listed in the registry, it's registered as:

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\HTML Help]

Name = myfile.chm

Value = "c:\chmfolder\"

This allows for opening the CHM without specifying the path, as in:

```
c:\windows\hh.exe myfile.chm
```

On Windows NT, this is generally seen as:

```
c:\winnt\hh.exe myfile.chm
```

Used with the CHMFile property, provides a method for specifying multiple HTML Help files within the class.

Note: This method does not exist in the HTML Help Object Library DLL.

Syntax

object.HHSetHelpFile(*integer*)

The HHDefaultURL property syntax has these parts:

object An object expression that evaluates to an instance of the HTMLHelp object.

integer An integer specifying which of any number of HTML Help files to use.

Example

```
Private hHelp as new HTMLHelp
Private Sub mnuHelpContents_Click()

    ' Display the Contents tab of the Help file
    With hHelp
        .CHMFile = .HHSetHelpFile(1)
        .HHWindow = "Main"
        .HHDisplayContents Me.hwnd
    End With

End Sub
```

Comments

The actual code within the class for this function is as follows:

```
Public Function HHSetHelpFile(ByVal intSelHelpFile As Integer) As String

    ' Set the string variable to
    ' include the application path
    Select Case intSelHelpFile
    Case 1
        HHSetHelpFile = App.Path & "\Library.chm"
    Case 2
        ' list other HTML Help files here
    End Select

End Function
```

This is supplied in order to provide a method for specifying multiple HTML Help files within the class. The line:

```
HHSetHelpFile = App.Path & "\Library.chm"
```

needs to be replaced with a line containing the name of an HTML Help file to be used with an application. The Select...Case block can then be expanded to include as many HTML Help files as necessary to the limit of the variable intSelHelpFile. This is particularly useful when more than one Help file is shipped, and is even more useful when it's necessary to call any single Help file from

within a set of merged HTML Help files.

For more information on merging HTML Help files, see Robert Chandler's information on the subject at http://www.helpware.net/htmlhelp/how_to_merge.htm.

Deletes the specified CHM file from the registry. This method calls the HHCheckRegistry method at the end of its routine as verification of the registration process. If the specified file is not registered before HHUnRegister is called, an error message is generated.

Syntax

object.**HHUnRegister** (*file as string*)

object An object expression that evaluates to an instance of the HTMLHelp object.

file The CHM file, including the path, to unregister from the registry.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileUnRegister_Click()
    Dim strFileToUnRegister As String
    strFileToUnRegister = txtFileName.Text

    With hHelp
        .HHUnRegister strFileToUnRegister
        Debug.Print .HHRegFileName & " was registered as:"
        Debug.Print .HHRegFilePath
    End With
End Sub
```

Remarks

When an HTML Help file is listed in the registry, it's registered as:

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\HTML Help]

Name = myfile.chm

Value = "c:\chmfolder\"

This allows for opening the CHM without specifying the path, as in:

```
c:\windows\hh.exe myfile.chm
```

On Windows NT, this is generally seen as:

```
c:\winnt\hh.exe myfile.chm
```

Returns True if the HTML Help and Internet Explorer versions on the system are higher than what's desired, False if they're not.

Syntax

object.**HHVerifyMinConfig** (*desired HH version*, *desired IE version*)

object An object expression that evaluates to an instance of the HTMLHelp object.

desired HH version One of the following constants:

HH_1_0
HH_1_1
HH_1_1A
HH_1_1B
HH_1_2
HH_1_21
HH_1_21A
HH_1_22

desired IE version One of the following constants:

IE_3_0
IE_3_0_OSR2
IE_3_01
IE_3_02
IE_4_0_PP2
IE_4_0
IE_4_01
IE_4_01_SP1
IE_4_01_SP2
IE_5_0_Beta1
IE_5_0_Beta2
IE_5_0
IE_5_0A
IE_5_0B

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpVerify_Click()
    MsgBox "System up-to-date: " & _
        hHelp.HHVerifyMinConfig(HH_1_1, IE_5_0B) & Chr(13) & _
        "HTML Help Version: " & hHelp.HHVersion & Chr(13) & _
        "Internet Explorer Version: " & hHelp.IEVersion, _
        vbOKOnly + vbInformation, "Version Information"
End Sub
```

Returns **True** if HTML Help is installed on the system, **False** if it is not.

Syntax

boolVariable = **HHInstalled**

The **HHInstalled** property syntax has these parts:

Part	Description
<i>boolVariable</i>	A Boolean variable to contain the return value of the HHInstalled property.

Example

```
Private hHelp As New HTMLHelp
MsgBox "HTML Help Installed: " & hHelp.HHInstalled, _
    vbOKOnly + vbInformation, "Installation Status"
```

Remarks

This property is provided for those developers who wish to determine whether or not HTML Help is installed.

The DLL determines whether or not HTML Help is installed by checking the registry for the existence of a proper value. This was coded according to a suggestion made to me by Ralph Walden, lead developer of HTML Help at Microsoft until March of 1998.

For Visual Basic 6.0 and above: If you've created an HTML Help file for your application and set the application's **HelpFile** property to the CHM file, Visual Basic automatically calls Help when a user presses the F1 key. If there is a context number in the **HelpContextID** property for either the active control or the active form, Help displays a topic corresponding to the current Help context; otherwise it displays the main contents screen.

Specifies the path and filename of an HTML Help file used by your application to display Help or online documentation.

Syntax

object.CHMFile[= *filename*]

The **CHMFile** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>filename</i>	A string expression specifying the path and filename of the HTML Help file for your application.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpContents_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHWindow = "Main"
        .HHDisplayContents
    End With

End Sub
```

Remarks

For Visual Basic 6.0 and above: If you've created an HTML Help file for your application and set the application's **HelpFile** property to the CHM file, Visual Basic automatically calls Help when a user presses the F1 key. If there is a context number in the **HelpContextID** property for either the active control or the active form, Help displays a topic corresponding to the current Help context; otherwise it displays the main contents screen.

Returns the current version of Internet Explorer on the system as a String expression. Read only.

Syntax

object.IEVersion

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpVerify_Click()
    MsgBox "System up-to-date: " & _
        hHelp.HHVerifyMinConfig(HH_1_1, IE_5_0B) & Chr(13) & _
        "HTML Help Version: " & hHelp.HHVersion & Chr(13) & _
        "HTML Help Friendly Name: " & hHelp.HHFriendlyName & Chr(13) & _
        "Internet Explorer Version: " & hHelp.IEVersion & Chr(13) & _
        "Internet Explorer Friendly Name: " & hHelp.IEFriendlyName, _
        vbOKOnly + vbInformation, "Version Information"
End Sub
```

Remarks

The version information is extracted from iexplore.exe.

Specifies a keyword to search for using HHD`DisplayALink`.

Syntax

`object.HHALink[= ALink keywords]`

The `HHALink` property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the <code>HTMLHelp</code> object.
<i>ALink keywords</i>	A string expression specifying the keyword to search for using <code>HHD<code>DisplayALink</code></code> . It can also be a list of keywords, separated by semicolons.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdALink_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicNotFound.txt
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = "This is where the title would go."
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHALink = "chmfile;copyright"
        .HHDDisplayALink
    End With

End Sub
```

Specifies the URL to use if a keyword cannot be found.

Syntax

object.HHDefaultURL[= HTML Help URL]

The HHDefaultURL property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>HTML Help URL</i>	A string expression specifying the URL of a "topic not found" topic in the HTML Help file specified in the HHWindow property when used with the HHDisplayALink and HHDisplayKeyword methods. Can be set to vbNullString if the HHMsgText and HHMsgTitle properties are to be used instead.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdKeyword_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = "html\_topic_not_found.htm"
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = txtTitle.Text
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHKeyWord = txtKeyword.Text
        .HHDisplayKeyword
    End With

End Sub
```

Returns the friendly name of the HTML Help version as a String expression (i.e, "1.21a").

Syntax

object.HHFriendlyName

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpVerify_Click()
    MsgBox "System up-to-date: " & _
        hHelp.HHVerifyMinConfig(HH_1_1, IE_5_0B) & Chr(13) & _
        "HTML Help Version: " & hHelp.HHVersion & Chr(13) & _
        "HTML Help Friendly Name: " & hHelp.HHFriendlyName & Chr(13) & _
        "Internet Explorer Version: " & hHelp.IEVersion & Chr(13) & _
        "Internet Explorer Friendly Name: " & hHelp.IEFriendlyName, _
        vbOKOnly + vbInformation, "Version Information"
End Sub
```

Remarks

The friendly name is determined from version information extracted from hhctrl.ocx.

Returns the friendly name of the Internet Explorer version as a String expression (i.e, "5.0").

Syntax

object.IEFriendlyName

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpVerify_Click()
    MsgBox "System up-to-date: " & _
        hHelp.HHVerifyMinConfig(HH_1_1, IE_5_0B) & Chr(13) & _
        "HTML Help Version: " & hHelp.HHVersion & Chr(13) & _
        "HTML Help Friendly Name: " & hHelp.HHFriendlyName & Chr(13) & _
        "Internet Explorer Version: " & hHelp.IEVersion & Chr(13) & _
        "Internet Explorer Friendly Name: " & hHelp.IEFriendlyName, _
        vbOKOnly + vbInformation, "Version Information"
End Sub
```

Remarks

The friendly name is determined from version information extracted from iexplore.exe.

Specifies the text to display in a message box if a keyword cannot be found.

Syntax

object.HHMsgText[= *message text*]

The HHMsgText property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>message text</i>	A string expression specifying the text to display within a standard message box. Can be set to vbNullString if the HHDefaultURL property is to be used instead.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdKeyword_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicNotFound.txt
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = txtTitle.Text
            .HHMsgText = "This is where the text would go."
        End If
        .HHWindow = "Main"
        .HHKeyWord = txtKeyword.Text
        .HHDisplayKeyword
    End With

End Sub
```

Specifies the title to display on a message box if a keyword cannot be found.

Syntax

object.HHMsgTitle[= *message title*]

The HHMsgTitle property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>message title</i>	A string expression specifying the title to display on a standard message box. Can be set to vbNullString if the HHDefaultURL property is to be used instead.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdKeyword_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicNotFound.txt
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = "This is where the title would go."
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHKeyWord = txtKeyword.Text
        .HHDisplayKeyword
    End With

End Sub
```

Sets the back color of a popup generated with `HHDisplayPopup` using the standard VB color constants. `HHPopupCustomColors` must be set to `False`.

Syntax

`object.HHPopupBackColor [= colorconstant]`

The `HHPopupBackColor` property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the <code>HTMLHelp</code> object.
<i>colorconstant</i>	One of the following Visual Basic color constants: vbBlack (&H000000) vbBlue (&HFF0000) vbCyan (&HFFFF00) vbGreen (&HFF00) vbMagenta (&HFF00FF) vbRed (&HFF) vbWhite (&HFFFFFF) vbYellow (&HFFFF)

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupCustomColors = False  
        .HHPopupBackColor = vbBlack  
        .HHPopupTextColor = vbWhite  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Sets the back color of a popup generated with `HHDisplayPopup` using custom colors in the form `&HBBGGRR`. `HHPopupCustomColors` must be set to `True`.

Syntax

```
object.HHPopupCustomBackColor[ = customcolor]
```

The `HHPopupCustomBackColor` property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the <code>HTMLHelp</code> object.
<i>customcolor</i>	A custom color in the form <code>&HBBGGRR</code>

Comments

To set the colors and text to match those from an `HH_TP_HELP_WM_HELP` popup:

```
.HHPopupCustomColors = True  
.HHPopupCustomBackColor = &HFFFF  
.HHPopupCustomTextColor = &HFFFF  
.HHPopupTextSize = "8"
```

Example

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFC0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

True of the HHPopupCustomTextColor and HHPopupCustomBackColor are going to be used, False if not.

Syntax

object.HHPopupCustomColors[= *boolean*]

The HHPopupCustomColors property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>boolean</i>	Set to True of the HHPopupCustomTextColor and HHPopupCustomBackColor are going to be used, False if not.

Comments

To set the colors and text to match those from an HH_TP_HELP_WM_HELP popup:

```
.HHPopupCustomColors = True  
.HHPopupCustomBackColor = &HFFFF  
.HHPopupCustomTextColor = &HFFFF  
.HHPopupTextSize = "8"
```

Example

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFFFC0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Sets the text color for a popup generated with `HHDisplayPopup` using custom colors in the form `&HBBGGRR`. `HHPopupCustomColors` must be set to `True`.

Syntax

`object.HHPopupCustomTextColor`[= *customcolor*]

The `HHPopupCustomTextColor` property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the <code>HTMLHelp</code> object.
<i>customcolor</i>	A custom color in the form <code>&HBBGGRR</code>

Comments

To set the colors and text to match those from an `HH_TP_HELP_WM_HELP` popup:

```
.HHPopupCustomColors = True  
.HHPopupCustomBackColor = &FFFFFF  
.HHPopupCustomTextColor = &FFFFFF  
.HHPopupTextSize = "8"
```

Example

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFFFC0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Specifies the text file containing the popup text as listed in the [TEXT POPUPS] section of a CHM. The popup is generated with HHDisplayPopup.

Syntax

object.HHPopupFile[= *filename*]

The **HHPopupFile** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>filename</i>	A string expression specifying the name of a popup text file.

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Specifies the resource containing the to retrieve popup text from as listed in a Visual Basic project or the context integer of a popup topic as specified in a valid popup file in a CHM. The popup is generated with HHDisplayPopup.

Syntax

object.HHPopupID[= *id*]

The HHPopupID property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>id</i>	The resource containing the to retrieve popup text from as listed in a Visual Basic project or the context integer of a popup topic as specified in a valid popup file in a CHM.

Examples

For use with a CHM file:

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

For use with a resource file:

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFFFC0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Specifies the text to display in a simple text popup. The popup is generated with HHDisplayPopup.

Syntax

object.HHPopupText[= *string*]

The HHPopupText property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>message title</i>	A string expression specifying the text to display on the popup.

Example

In this example, note the use of Chr(10) to create multiple paragraphs in the popup:

```
Private Sub mnuExamplesTextPopup_Click()  
    Dim strPopup As String  
    With hHelp  
        .HHPopupType = HH_TEXT_POPUP  
        .HHPopupCustomColors = False  
        strPopup = "This is a test using a simple text string." & _  
            Chr(10) & Chr(10) & _  
            "Notice how you can have more than one paragraph."  
        .HHPopupText = strPopup  
        .HHDisplayPopup Me.hWnd  
    End With  
End Sub
```

Set to True to make the popup text **bold**, False otherwise. The popup is generated with HHPopupType and HHDisplayPopup.

Syntax

object.HHPopupTextBold[= *boolean*]

The HHPopupTextBold property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>boolean</i>	Set to True to make the popup text bold , False otherwise.

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupTextBold = True  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Sets the text color for a popup generated with HHDisplayPopup using the standard VB color constants. HHPopupCustomColors must be set to False.

Syntax

object.HHPopupTextColor[= *colorconstant*]

The HHPopupTextColor property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>colorconstant</i>	One of the following Visual Basic color constants: vbBlack (&H000000) vbBlue (&HFF0000) vbCyan (&HFFFF00) vbGreen (&HFF00) vbMagenta (&HFF00FF) vbRed (&HFF) vbWhite (&HFFFFFF) vbYellow (&HFFFF)

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupCustomColors = False  
        .HHPopupBackColor = vbBlack  
        .HHPopupTextColor = vbWhite  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Name of the font to be used in a text popup. The popup is generated with `HHDisplayPopup`.

Syntax

`object.HHPopupTextFont[= string]`

The `HHPopupTextFont` property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the <code>HTMLHelp</code> object.
<i>string</i>	A string expression specifying the name of the font to be used in a popup. The font must be a valid font on the user's system or the default popup font will be used.

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupTextFont = "Comic Sans MS"  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Set to True to make the popup text *italicized*, False otherwise. The popup is generated with HHDisplayPopup.

Syntax

`object.HHPopupTextItalic[= boolean]`

The **HHPopupTextItalic** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>boolean</i>	Set to True to make the popup text <i>italicized</i> , False otherwise.

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupTextItalic = True  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Point size of the font used in a text popup. The popup is generated with HHDisplayPopup.

Syntax

object.HHPopupTextSize[= *string*]

The HHPopupTextSize property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>string</i>	A string expression specifying a valid point size for the selected font for the popup.

Comments

To set the colors and text to match those from an HH_TP_HELP_WM_HELP popup:

```
.HHPopupCustomColors = True  
.HHPopupCustomBackColor = &HFFFFFF  
.HHPopupCustomTextColor = &HFFFFFF  
.HHPopupTextSize = "8"
```

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupTextSize = "8"  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

Set to True to make the popup text underlined, False otherwise. The popup is generated with HHDisplayPopup.

Syntax

object.HHPopupTextUnderline[= *boolean*]

The HHPopupTextUnderline property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>boolean</i>	Set to True to make the popup text <u>underlined</u> , False otherwise.

Example

```
Private Sub mnuExamplesCHMPopup_Click()  
    With hHelp  
        .HHPopupType = HH_CHM_POPUP  
        .CHMFile = .HHSetHelpFile(1)  
        .HHPopupFile = .HHSetHelpFile(2)  
        .HHPopupID = 21000  
        .HHPopupTextUnderline = True  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

The type of popup to be generated by HHDisplayPopup.

Syntax

object.HHPopupType[= *constant*]

The HHPopupType property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>constant</i>	One of the following three constants: HH_CHM_POPUP HH_RESOURCE_POPUP HH_TEXT_POPUP

Examples

The first example illustrates how to create a simple popup from a string containing the desired information:

```
Private Sub mnuExamplesTextPopup_Click()  
    With hHelp  
        .HHPopupType = HH_TEXT_POPUP  
        .HHPopupCustomColors = False  
        .HHPopupText = "This is a test using a simple text string."  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

The next example demonstrates getting the text for the popup from a resource file. In this case, the resource ID is 101. This also demonstrates the use of custom colors for the popup text and background. These colors are in the form &HBBGRR.

```
Private Sub mnuExamplesResourcePopup_Click()  
    With hHelp  
        .HHPopupType = HH_RESOURCE_POPUP  
        .HHPopupID = 101  
        .HHPopupCustomColors = True  
        .HHPopupCustomBackColor = &H808000  
        .HHPopupCustomTextColor = &HFFFFFFC0  
        .HHDisplayPopup Me.hwnd  
    End With  
End Sub
```

The final example illustrates providing the popup for the text from the popup topics in a CHM file. Also shown are the use of standard colors for the popup's text and background, and a custom font.

```
Private Sub mnuExamplesCHMPopup_Click()
```

```

With hHelp
    .HHPopupType = HH_CHM_POPUP
    .CHMFile = .HHSetHelpFile(1)
    .HHPopupFile = .HHSetHelpFile(2)
    .HHPopupID = 21000
    .HHPopupCustomColors = False
    .HHPopupBackColor = vbBlack
    .HHPopupTextColor = vbWhite
    .HHPopupTextFont = "Comic Sans MS"
    .HHPopupTextSize = "8"
    .HHPopupTextBold = True
    .HHPopupTextItalic = True
    .HHPopupTextUnderline = True
    .HHDisplayPopup Me.hwnd
End With
End Sub

```

In this situation, HHSetHelpFile is modified to also contain the name of the popup file compiled into the CHM:

```

Public Function HHSetHelpFile(ByVal intSelHelpFile As Integer) As String
    ' Set the string variable to
    ' include the application path
    Select Case intSelHelpFile
    Case 1
        HHSetHelpFile = App.Path & "\Library.chm"
    Case 2
        ' Popup text file for the above CHM
        HHSetHelpFile = "Library.txt"
    Case Else
        ' list other HTML Help files here
    End Select
End Function

```

Set by the HHCheckRegistry method. HHCheckRegistry checks the registration of the specified CHM file. If the file is found to be registered, the HHRegFileName and HHRegFilePath properties are loaded with the found information. Once the information is collected from the registry, the file is located. If it's in the registered location, HHRegFileExists is set to True. Otherwise, HHRegFileExists is set to False. If the file is not registered, the HHRegFileName and HHRegFilePath properties return empty strings.

Syntax

object.**HHRegFileExists** (=boolean)

object An object expression that evaluates to an instance of the HTMLHelp object.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileCheckRegistry_Click()
    Dim strFileToCheck As String
    strFileToCheck = txtFileName.Text
    hHelp.HHCheckRegistry strFileToCheck
    If hHelp.HHRegFileName = "" Then
        MsgBox strFileToCheck & " is not registered.", _
            vbExclamation + vbOKOnly, _
            "Registry Check Result"
    Else
        MsgBox strFileToCheck & " is registered as " & _
            & hHelp.HHRegFilePath & Chr(10) & _
            "File exists: " & hHelp.HHRegFileExists, _
            vbInformation + vbOKOnly, _
            "Registry Check Result"
    End If
End Sub
```

Remarks

When an HTML Help file is listed in the registry, it's registered as:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\HTML Help]
```

```
Name = myfile.chm
```

```
Value = "c:\chmfolder\"
```

This allows for opening the CHM without specifying the path, as in:

```
hhhelp.CHMFile = "myfile.chm"
```

and

```
c:\windows\hh.exe myfile.chm
```

On Windows NT, this is generally seen as:

c:\winnt\hh.exe myfile.chm

Used in conjunction with HHCheckRegistry, returns the confirmed CHM file name. Read only.

Syntax

object.HHRegFileName[= *filename*]

The HHRegFileName property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>filename</i>	The name of the confirmed CHM file if found.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileCheckRegistry_Click()
    Dim strFileToCheck As String
    strFileToCheck = txtFileName.Text
    hHelp.HHCheckRegistry strFileToCheck
    If hHelp.HHRegFileName = "" Then
        MsgBox strFileToCheck & " is not registered.", _
            vbExclamation + vbOKOnly, _
            "Registry Check Result"
    Else
        MsgBox strFileToCheck & " is registered as " & _
            & hHelp.HHRegFilePath & Chr(10) & _
            "File exists: " & hHelp.HHRegFileExists, _
            vbInformation + vbOKOnly, _
            "Registry Check Result"
    End If
End Sub
```

Used in conjunction with HHCheckRegistry, returns the path of the confirmed CHM file name. Read only.

Syntax

object.HHRegFilePath[= *filepath*]

The HHRegFilePath property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>filepath</i>	The path of the confirmed CHM file if found.

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuFileCheckRegistry_Click()
    Dim strFileToCheck As String
    strFileToCheck = txtFileName.Text
    hHelp.HHCheckRegistry strFileToCheck
    If hHelp.HHRegFileName = "" Then
        MsgBox strFileToCheck & " is not registered.", _
            vbExclamation + vbOKOnly, _
            "Registry Check Result"
    Else
        MsgBox strFileToCheck & " is registered as " & _
            & hHelp.HHRegFilePath & Chr(10) & _
            "File exists: " & hHelp.HHRegFileExists, _
            vbInformation + vbOKOnly, _
            "Registry Check Result"
    End If
End Sub
```

Specifies a keyword to search for using HHDisplayKeyword.

Syntax

object.**HHKeyword**[= *keywords*]

The **HHKeyword** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>keywords</i>	A string expression specifying the keyword to search for using HHDisplayKeyword. It can also be a list of keywords, separated by semicolons.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdKeyword_Click()

    With hHelp
        .CHMFile = App.Path & "\helpdll.chm"
        If optTopic = True Then
            .HHDefaultURL = txtTopicNotFound.txt
            .HHMsgTitle = vbNullString
            .HHMsgText = vbNullString
        Else
            .HHDefaultURL = vbNullString
            .HHMsgTitle = "This is where the title would go."
            .HHMsgText = txtText.Text
        End If
        .HHWindow = "Main"
        .HHKeyword = "chmfile;copyright"
        .HHDisplayKeyword
    End With

End Sub
```

If set to True, the HTML Help window will be set as a sibling of the calling window.

Syntax

object.**HHShowOnTop**[= *boolean*]

The **HHShowOnTop** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>boolean</i>	True to set the HTML Help viewer as a sibling, False to keep it independent.

Example

```
Private hHelp As New HTMLHelp

Private Sub mnuHelpContents_Click()

    With hHelp
        .CHMFile = txtFileName.Text
        .HHWindow = "main"

        If chkShowOnTop Then
            .HHShowOnTop = True
        Else
            .HHShowOnTop = False
        End If

        .HHDisplayContents Me.hWnd
    End With

End Sub
```

These subroutine does not reside in the HTML Help class, but is located in modHHSUBClass.

Note: These procedures do not exist in the HTML Help Object Library DLL.

The intent of the HHSUBclass subroutine is to allow for the subclassing of the calling form in order to enable the interception of Windows messaging for the form. This will allow Visual Basic 5 developers to use HTML Help-based What's This Help, and provides a method for all Visual Basic developers to use customized What's This popups. With a bit of work, it also allows for the separation of What's This Help and the operation of the F1 key. This is shown in code in the usage section below.

PLEASE NOTE:

Due to the use of the AddressOf operator, the HHSUBclass subroutine **will** crash the Visual Basic IDE in Debug mode. If debugging of the application is necessary, uncomment the first End Sub in the HHSUBclass subroutine.

Syntax

HHSUBclass (form)

The **HHTopicID** property syntax has these parts:

Part	Description
<i>form</i>	The name of the calling form. An hWnd will not work in this case.

Example

```
Private Sub Form_Load()  
    HHSUBclass Me  
End Sub
```

When a subclassed form is closed, it is important to make sure any popups are closed. This can be done by setting the focus to the form in the QueryUnload event. Then, make sure you unsubclass the form:

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)  
    ' Close any open popups  
    Me.SetFocus  
  
    ' Clean up the HH class  
    hHelp.HHClose  
    Set hHelp = Nothing  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    HHUnSUBClass Me  
End Sub
```

Usage

The calling form needs have the following procedures added to its module:

- Public Sub OnContextMenu(hWndControl As Long)
- Public Sub OnHelp(hWndControl As Long)
- Public Sub OnNavComplete(phhnt As Long)
- Public Sub OnTCard(wParam As Long, lParam As Long)
- Public Sub OnTrack(phhnn As Long)
- Public Sub OnWindowCreate(phhnt As Long)

The following example for the OnHelp procedure allows for the displaying of a Help topic if the F1 key is pressed, or a popup topic if the What's This button is clicked. This is set up for the same form and accurately differentiates between the two functions. As the example uses the custom popup techniques in the HTML Help class file, the text for the popup can be derived from a text string, a resource file, or a CHM file. The latter is shown for completeness. In this example, the Help topics are called via calling the procedure for a command button on the form, and the What's This Help topic ID's are stored in the Tag properties of the associated controls. Also, a generic popup is used if the Tag property is empty, and uses multiple paragraphs in its text message.

```
Private Const VK_F1 = &H70
Private Const VK_NUMLOCK = &H90
Private Const VK_CAPITAL = &H14
Private Const VK_SCROLL = &H91

Private Type KeyboardBytes
    kbByte(0 To 255) As Byte
End Type

Private kbArray As KeyboardBytes

Private Declare Function GetKeyState Lib "user32" _
    (ByVal nVirtKey As Long) As Integer

Private Declare Function SetKeyboardState Lib "user32" _
    (kbArray As KeyboardBytes) As Long

Public Sub OnHelp(hWndControl As Long)

    ' This procedure allows for the displaying of a Help
    ' topic if the F1 key is pressed, or a popup topic if
    ' the What's This button is clicked

    On Error GoTo ErrHandler

    Dim frmCurrent As Form
    Dim ctlControl As Control
    Dim errNumber As Integer
    Dim intF1Key As Integer
    Dim intNumLockKey As Integer
    Dim intCapLockKey As Integer
    Dim intScrollLockKey As Integer
    Dim lngScan As Long
    Dim strCustom As String
```

```

' Get the latest state of the F1 key
intF1Key = GetKeyState(VK_F1)

' If F1 has been pressed, call the Help topic
If (intF1Key = -127) Or (intF1Key = -128) Then

    ' Call the Help topic
    cmdHelp_Click

    GoTo ExitRoutine

End If

' If F1 has not been pressed, display a popup by
' looping through all the controls, finding the one
' that matches the WM_HELP message parameter, and
' using the info in the Tag property for the context ID
For Each frmCurrent In Forms
    For Each ctlControl In frmCurrent.Controls
        If (ctlControl.hWnd = hWndControl) Then
            With hHelp

                ' Set the popup type to CHM-based
                .HHPopupType = HH_CHM_POPUP

                ' Specify the CHM file and the internal
                ' text popup file
                .CHMFile = .HHSetHelpFile(1)
                .HHPopupFile = .HHSetHelpFile(2)

                ' Set the colors and text to match those
                ' from a HH_TP_HELP_WM_HELP popup
                .HHPopupCustomColors = True
                .HHPopupCustomBackColor = &HFFFFFF
                .HHPopupCustomTextColor = &HFFFFFF
                .HHPopupTextSize = "8"

                ' Get the context integer from the Tag property
                .HHPopupID = CLng(ctlControl.Tag)

                ' If the control has a Tag property, we're ok
                If errNumber <> 438 Then
                    ' Tag property is empty

                    If ctlControl.Tag = "" Then
                        ' No topic ID is specified in the Tag property,
                        ' so send up a generic message in order to
                        ' prevent HH error message.
                        strCustom = "HHSubclass message:" & _
                            Chr(10) & _
                            "You need to yell at the Help author for " & _
                            "not creating a Help topic for this item!"
                        .HHPopupText = strCustom
                        .HHPopupType = HH_TEXT_POPUP
                        .HHDisplayPopup Me.hWnd
                    End If
                End If
            End With
        End If
    End For
End For

```

```

        Else
            ' Display the specified CHM-based popup topic
            .HHDisplayPopup Me.hWnd
            errNumber = 0
        End If

        ' If the control doesn't have a Tag property,
        ' deal with it.
        Else
            errNumber = 0
        End If

    End With
End If

Next ctlControl
Next frmCurrent

ExitRoutine:

' Clear any previous F1 keypress,
' but leave the Lock keys alone
intNumLockKey = GetKeyState(VK_NUMLOCK)
intCapLockKey = GetKeyState(VK_CAPITAL)
intScrollLockKey = GetKeyState(VK_SCROLL)
With kbArray
    .kbByte(VK_F1) = 0
    .kbByte(VK_NUMLOCK) = intNumLockKey
    .kbByte(VK_CAPITAL) = intCapLockKey
    .kbByte(VK_SCROLL) = intScrollLockKey
End With
SetKeyboardState kbArray

Exit Sub

ErrorHandler:

Select Case Err.Number
Case 438
    ' The control being checked doesn't have a
    ' Tag property, so set the variable to Err.Number
    ' before it gets cleared
    errNumber = Err.Number
    Resume Next

Case Else
    ' Nothing else to really be concerned with
    Resume Next

End Select

End Sub

```

Specifies a context integer for the topic to display, as defined in the [MAP] section of the HTML Help project file.

Syntax

object.**HHTopicID**[= *integer*]

The **HHTopicID** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>integer</i>	A context integer for the topic to display, as defined in the [MAP] section of the HTML Help project file.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicID = 1030
        .HHWindow = "Main"
        .HHDisplayTopicID
    End With

End Sub
```

Remarks

For Visual Basic 6.0 and above: If you've created an HTML Help file for your application and set the application's **HelpFile** property to the CHM file, Visual Basic automatically calls Help when a user presses the F1 key. If there is a context number in the **HelpContextID** property for either the active control or the active form, Help displays a topic corresponding to the current Help context; otherwise it displays the main contents screen.

Specifies a URL for the topic to display, as defined in the [FILES] section of the HTML Help project file.

Syntax

object.HHTopicURL[= *URL*]

The HHTopicURL property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>URL</i>	A URL for the topic to display, as defined in the [FILES] section of the HTML Help project file.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicURL = "html\intro.htm"
        .HHWindow = "Main"
        .HHDisplayTopicURL
    End With

End Sub
```

Remarks

For Visual Basic 6.0 and above: If you've created an HTML Help file for your application and set the application's **HelpFile** property to the CHM file, Visual Basic automatically calls Help when a user presses the F1 key. If there is a context number in the **HelpContextID** property for either the active control or the active form, Help displays a topic corresponding to the current Help context; otherwise it displays the main contents screen.

Returns the current version of HTML Help on the system as a String expression. Read only.

Syntax

object.HHVersion

Example

```
Private hHelp As New HTMLHelp
Private Sub mnuHelpVerify_Click()
    MsgBox "System up-to-date: " & _
        hHelp.HHVerifyMinConfig(HH_1_1, IE_5_0B) & Chr(13) & _
        "HTML Help Version: " & hHelp.HHVersion & Chr(13) & _
        "HTML Help Friendly Name: " & hHelp.HHFriendlyName & Chr(13) & _
        "Internet Explorer Version: " & hHelp.IEVersion & Chr(13) & _
        "Internet Explorer Friendly Name: " & hHelp.IEFriendlyName, _
        vbOKOnly + vbInformation, "Version Information"
End Sub
```

Remarks

The version information is extracted from hhctrl.ocx.

Specifies the window to display as defined in the HTML Help project file.

Syntax

object.**HHWindow**[= *windowname*]

The **HHWindow** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an instance of the HTMLHelp object.
<i>windowname</i>	A string expression specifying the window to display.

Example

```
Private hHelp As New HTMLHelp
Private Sub cmdHelp_Click()

    With hHelp
        .CHMFile = App.Path & "\myapp\myhelp.chm"
        .HHTopicID = 1030
        .HHWindow = "Main"
        .HHDisplayTopicID
    End With

End Sub
```

Remarks

For Visual Basic 6.0 and above: If you've created an HTML Help file for your application and set the application's **HelpFile** property to the CHM file, Visual Basic automatically calls Help when a user presses the F1 key. If there is a context number in the **HelpContextID** property for either the active control or the active form, Help displays a topic corresponding to the current Help context; otherwise it displays the main contents screen.

This class is intended to be used with Microsoft Visual Basic 5.0 and above.

The author/developer assumes no responsibility and accepts no risk for the use of this class in any application developed by any means. If anything goes wrong, you're on your own!

September 1999 - Released version 3.0d

Methods

- HHCheckRegistry - Checks the registration of the specified CHM file.
- HHClose - Closes all open HTML Help windows.
- HHDisplayALink - Display a topic with an ALink as specified in the HHALink property.
- HHDisplayContents - Forces the HTML Help file to display the Contents tab.
- HHDisplayIndex - Forces the HTML Help file to display the Index tab.
- HHDisplayKeyword - Display a topic with a keyword as given in the HHKeyword property.
- HHDisplayPopup - Displays a text popup at the current coordinates of the mouse pointer.
- HHDisplaySearch - Forces the HTML Help file to display the Search tab.
- HHDisplayTopicID - Displays a given topic according to the HHTopicID property.
- HHDisplayTopicURL - Displays a given topic according to the HHTopicURL property.
- HHInvokeWhatsThisHelp - Converts the mouse pointer to the What's This mouse pointer. When the left button is then clicked on a control, What's This is invoked for that control. Great for creating a What's This? menu item. **Note:** This method does not exist in the HTML Help Object Library DLL.
- HHRegister - Registers the specified CHM file.
- HHSetHelpFile - Used with the CHMFile property, provides a method for specifying multiple HTML Help files within the class. **Note:** This method does not exist in the HTML Help Object Library DLL.
- HHUnregister - Deletes the specified CHM file from the registry.
- HHVerifyMinConfig - Returns True if the HTML Help and Internet Explorer versions on the system are higher than what's desired, False if they're not.

Properties

- CHMFile - Path and file name of the HTML Help file to display.
- HHALink - Specifies an ALink keyword to search for using HHDisplayALink.
- HHDefaultURL - Specifies the URL to use if a keyword cannot be found.
- HHFriendlyName - Returns the friendly name of the HTML Help version as a String

expression (i.e, "1.21a").

- HHInstalled - Returns True if HTML Help is installed on a given system.
- HHKeyword - Specifies a keyword to search for using HHDDisplayKeyword
- HHMsgText - Specifies the text to display in a message box if a keyword cannot be found.
- HHMsgTitle - Specifies the title to display on a message box if a keyword cannot be found.
- HHPopupBackColor - Sets the back color of a popup generated with HHDDisplayPopup using the standard VB color constants.
- HHPopupCustomBackColor - Sets the back color of a popup generated with HHDDisplayPopup using custom colors in the form &HBBGRR.
- HHPopupCustomColors - True if the HHPopupCustomTextColor and HHPopupCustomBackColor are going to be used, False if not.
- HHPopupCustomTextColor - Sets the text color for a popup generated with HHDDisplayPopup using custom colors in the form &HBBGRR.
- HHPopupFile - Specifies the text file containing the popup text as listed in the [TEXT POPUPS] section of a CHM.
- HHPopupID - Specifies the resource containing the to retrieve popup text from as listed in a Visual Basic project or the context integer of a popup topic as specified in a valid popup file in a CHM.
- HHPopupText - Specifies the text to display in a simple text popup.
- HHPopupTextBold - Set to True to make the popup text **bold**, False otherwise.
- HHPopupTextColor - Sets the text color for a popup generated with HHDDisplayPopup using the standard VB color constants.
- HHPopupTextFont - Name of the font to be used in a text popup.
- HHPopupTextItalic - Set to True to make the popup text *italicized*, False otherwise.
- HHPopupTextSize - Point size of the font used in a text popup.
- HHPopupTextUnderline - Set to True to make the popup text underlined, False otherwise.
- HHPopupType - The type of popup to be generated by HHDDisplayPopup.
- HHRegFileExists - Used in conjunction with HHCheckRegistry, returns True if the file exists at the registered path, False if it doesn't.
- HHRegFileName - Used in conjunction with HHCheckRegistry, returns the confirmed CHM file name.
- HHRegFilePath - Used in conjunction with HHCheckRegistry, returns path of the confirmed CHM file name.
- HHSetHelpFile - Used with the CHMFile property, provides a method for specifying multiple HTML Help files within the class.
- HHShowOnTop - If set to True, the HTML Help window will be set as a sibling of the calling window.
- HHTopicID - Specifies the context integer for the HHDDisplayTopicID method

- HHTopicURL - Specifies the topic path and file name for the HHDdisplayTopicURL method
- HHVersion - Returns the current version of HTML Help on the system as a String expression. Read only.
- HHWindow - Specifies the window for use with the HHDdisplayTopicID and HHDdisplayTopicURL methods.
- IEFriendlyName - Returns the friendly name of the Internet Explorer version as a String expression (i.e, "5.0").
- IEVersion - Returns the current version of Internet Explorer on the system as a String expression. Read only.

Subroutines

Note: These procedures do not exist in the HTML Help Object Library DLL.

- HHS subclass - Allows for the subclassing of the calling form in order to enable the interception of Windows messaging for the form. This will allow Visual Basic 5 developers to use HTML Help-based What's This Help, and provides a method for all Visual Basic developers to use customized What's This popups. With a bit of work, it also allows for the separation of What's This Help and the operation of the F1 key.
- HHUnSubclass - Releases a previously subclassed form.

A WinHelp file is also included, explaining the various methods and properties of the class in greater detail.

HTML Help class for Microsoft Visual Basic
Version 3.0

This program, its tutorials, and related Help files, are ©1999 by Delmar Computing Services, Tipton, Michigan, USA. All rights reserved.

Developed by David Liske, Tipton, Michigan, USA
Microsoft HTML Help MVP
<http://www.vbexplorer.com/htmlhelp.asp>

Proof-of-concept testing and some HTML Help
API research provided by Robert Chandler,
The HelpWare Group, and Varian Corporation,
Melbourne, Vic, Australia
Microsoft HTML Help MVP 1999
<http://www.helpware.net>

Some registry functionality re-developed from
original code written by Dave Scarmozzino
<http://www.TheScarms.com>

Beta testing (July 1999):

Lani Hardage, MDL Information Systems, Inc.
Steve Hsu, TREEV, Inc.
John Hunt, Lotus Development Corporation
Shirley Kelly, Corbel, A SunGard Company
Valerie A. Lipow, Compuware Corporation
Leyden Martinez, Copextel, S.A., Cuba
Alejandro Sicilia, Copextel, S.A., Cuba

Further testing by Dana Cline,
Lucent Technologies, Boulder, Colorado
Microsoft HTML Help MVP 1998, 1999

Modifications of HTML Help version-checking and
IE version-checking code, and Windows NT/2000
problem-solving,
by Leonardo Presciuttini, Italy (August 1999)

GetLongPath_Legacy logic checks and code modifications,
and all-around support of a programming geek's
chocolate and strange snacking needs, by:
Marnella Liske, RN, BSN
University of Michigan Medical Center

Please send any performance or functionality modifications of these files to delmar@tc3net.com

