

## HTML Help

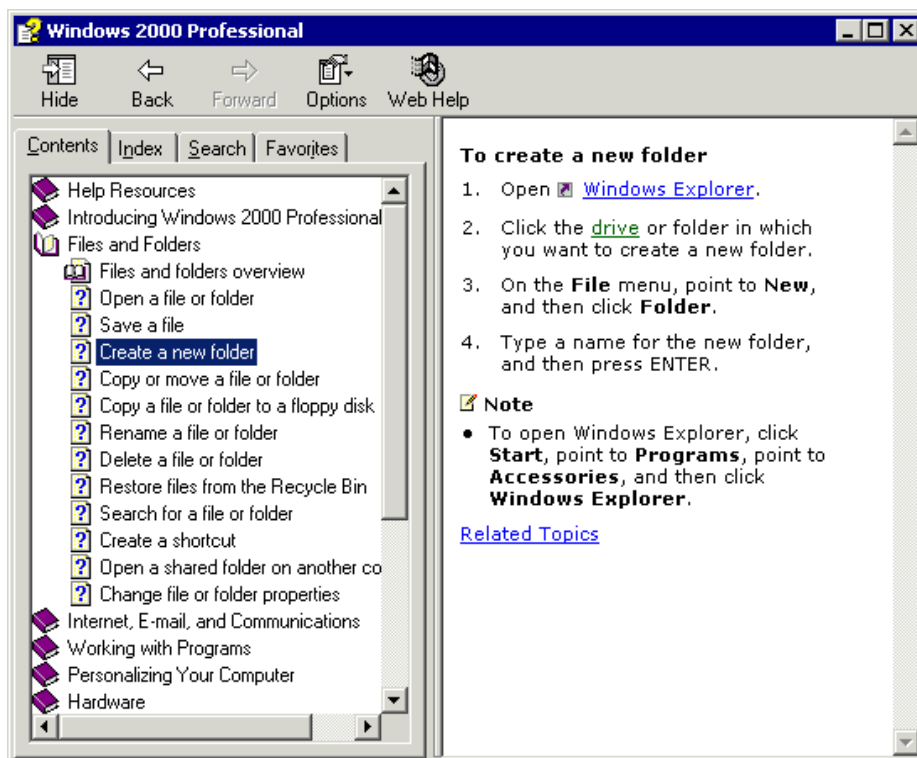
VB Web - <http://www.vbweb.co.uk/>

HTML Help is the new standard for Windows-based Help systems. It's fast becoming the Help system of choice, and is available to users of Internet Explorer 4.01 or greater and Windows 98. However, very little useful information is available from the MSDN CD for using it with Visual Basic. There is some information available from Microsoft, and if a developer is persistent through experimentation, some methods can be made to work. Other problems exist with attempting to call some of the C++ syntax used within an HTML Help project file from Visual Basic, in that the code simply will not recognize that particular syntax.

Parts of this article and code has been provided by David Liske of the [HTML Help Center](http://htmlhelpcenter.hypermart.net/). (<http://htmlhelpcenter.hypermart.net/>)

### What is HTML Help?

Below is a typical HTML Help window.



The main difference between HTML Help and WinHelp is that HTML Help displays both the topic and the contents at the same time. HTML Help also has a search feature, the files are HTML based, and therefore smaller than the RTF format files used in WinHelp. The final

compiled file is also smaller. This tutorial will be based around creating a HTML help file using the HTML Help Workshop, and then using the help in a test project. You can install the HTML Help Workshop, by running the setup program in the HTMLHelp directory of your VB CD (VB6 only). For more information you can visit the [HTML Workshop](#) at Microsoft.

### **BUG 1**

If you have the same version as I have of HTML Help Workshop, DO NOT click the Help tab in the main view, it will crash!

## **The Project File (1)**

There are six main parts to a HTML Project File:

<b>Options</b>	This section contains the basic information about the Help file.
<b>Windows</b>	This section contains information about the different Help Windows you are using
<b>Files</b>	This is simply a list of all the files in your HTML Help Project
<b>Alias</b>	Here, we're aliasing each required topic file to a topic ID. This will be explained later.
<b>Map</b>	This section assigns the values to each Alias, that we can use when we call the help file from VB
<b>Text Popup</b>	This file contains all of the text-based topics for What's This help

As an example, here is a (very small) HTML Project File:

```
[OPTIONS]
Auto Index=Yes
Compatibility=1.1
```

```
Compiled file=DevPadHelp.chm
Contents file=Contents.hhc
Default Window=Main
Default topic=html\welcome.htm
Display compile progress=Yes
Full-text search=Yes
Index file=Index.hhk
Language=0x809 English (United Kingdom)
Title=Developers Pad Help

[WINDOWS]
Main="Developers Pad 1 BETA 1
Help", "Contents.hhc", "Index.hhk", "html\
welcome.htm", "html\welcome.htm",,,,0x520,,0x206e,
[512,5,1153,496],,,,0,,,

[FILES]
html\welcome.htm
html\comments_suggestions.htm
html\copyright.htm
html\credits.htm
html\features.htm
html\report_bug.htm

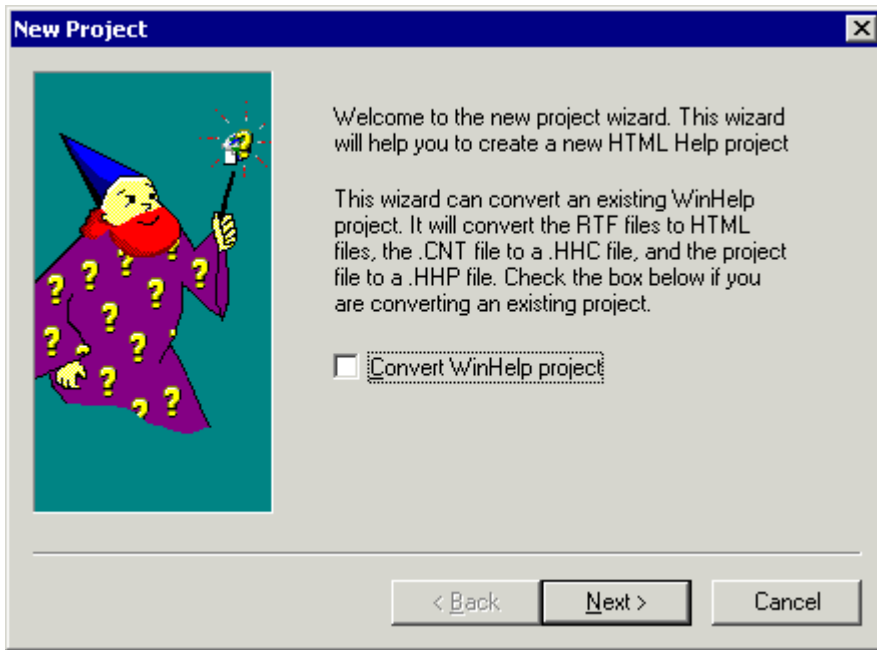
[ALIAS]
IDH_DLG_FIND=html\find_dialog.htm
IDH_DLG_INSERT_SYMBOLS=html\insert_symbols.htm
IDH_WELCOME=html\welcome.htm

[MAP]
#define IDH_DLG_FIND 1010
#define IDH_DLG_INSERT 1020
#define IDH_WELCOME 1000

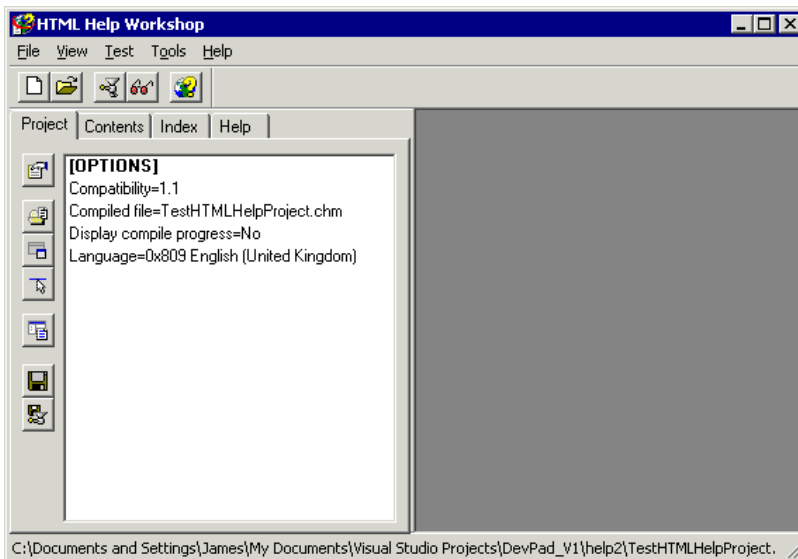
[TEXT POPUP]
popups.txt
```

First, lets create a new project. Run the HTML Help Workshop, and click New. Select, Project and click OK.


A wizard will then appear.

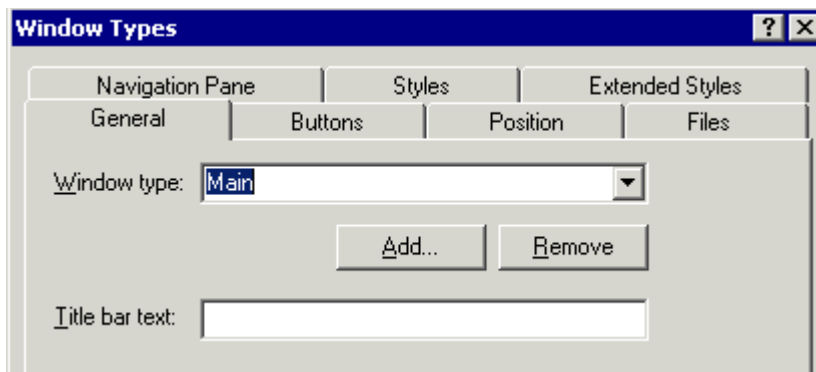


This gives you the option to convert a WinHelp project. We will ignore this option for the moment. Click next. Then, specify a location for your new project file. We will refer to this folder as the **root folder** from now on. Now, go into explorer, and create two subfolders in that folder named **html** and **images**. You are then given the option to specify any files you have already created. As we are going to start from scratch, don't specify any. Click Finish. This is the first stage completed. You will now see screen looking something like this:



## The Project File (2)

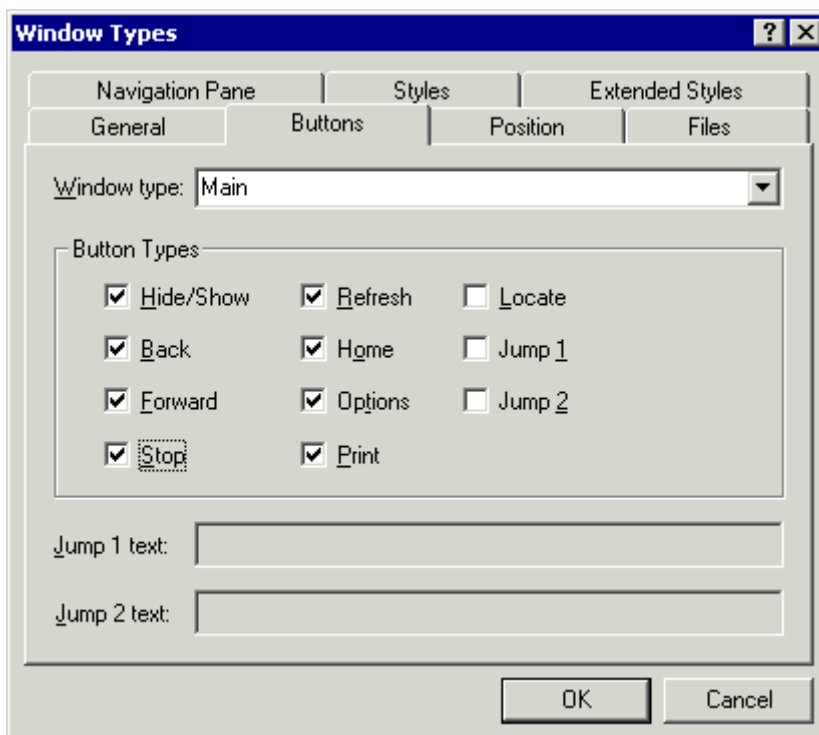
First, we need to create a window that will display our help file. Click the Add/Modify Window Definitions button . Enter Main into the dialog that is displayed, and click OK. Then the window below is displayed.



### BUG 2

If you have Version 1.1, you cannot remove a window by pressing the Remove button! You get a wonderful message informing you that 'This functionality has not yet been implemented'. To get around this, save and close the project. Then open the file in notepad, and delete the appropriate line under the [Windows] section.

The Window type is the name of the Window you have just created. Set the Title bar text to something like Notepad Help. Next, click the Buttons tab.



The buttons specified correspond to the buttons displayed across the top of the compiled Help file in the toolbar:



These buttons can be defined as follows:

- **Hide** - Hides the navigation Pane. When the pane is hidden, this becomes the

Show button.

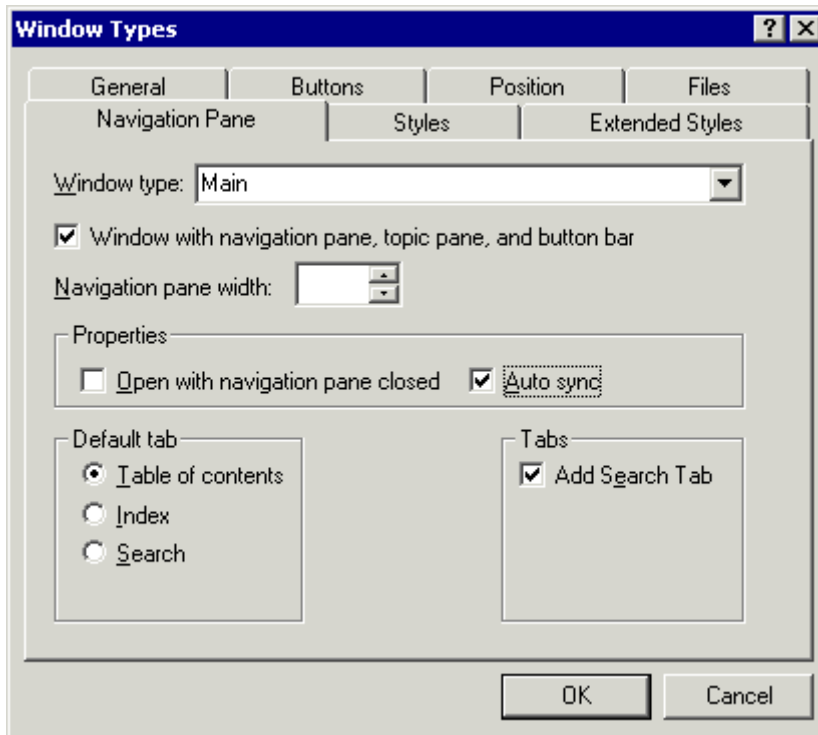
- **Back** - Move to the previously-viewed topic in this session.
- **Forward** - If Back has been used, moves to the next-viewed topic in this session.
- **Stop** - Stops the loading of any topic or web page in the viewer.
- **Refresh** - Refreshes the current page from the source.
- **Home** - Go to the page specified in the definition.
- **Print** - Prints the current page.
- **Options** - Presents the user with a menu of the other buttons.
- **Jump 1** - User-defined button.
- **Jump 2** - User-defined button.

Select the options you want. Next, click the Position tab. This allows you to specify the size and location of the help window. You can use the Autosizer to manually adjust where the window starts. For the tri-pane window, I normally set the width property to 640, and the height property to 440, making it slightly smaller than the standard 640 x 480 resolution, so the people with that setting don't experience problems. The Left and Top settings are at 0 so that the window is displayed in the corner of the screen.

The Files tab allows you to specify the files you will use in your project:


- **TOC** - File containing the entries for the Contents tab (HHC extension).
- **Index** - File containing the entries for the Index tab (HHK extension).
- **Default** - The topic to be displayed when the window first opens (HTML extension).
- **Home** - The topic HTMLHelp displays when the Home button is clicked (HTML extension).
- **Jump 1** - The topic HTMLHelp displays when the Jump 1 button is clicked.
- **Jump 2** - The topic HTMLHelp displays when the Jump 2 button is clicked.

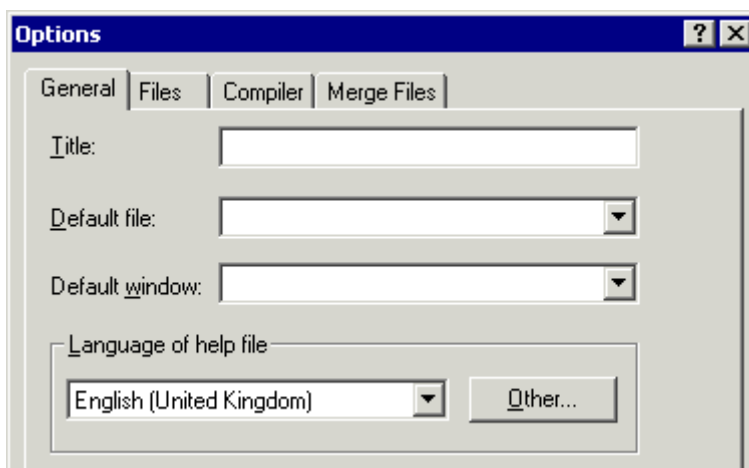
Don't worry about any of these at the moment, we will specify them later. Finally the Navigation Pane:



This tab allows you to specify what tabs are shown in the help window, and if it is a tri-pane (ie with Toolbar, nav pane and topic pane). Check the Window with navigation pane box, and ensure that Table of contents is selected in the Default tab frame. This specifies that the Table of contents will be the tab that is selected when the window is first displayed.

Also check the Auto sync check box. This ensures that when the window is displayed, the Table of contents is displaying the current topic. You can also check the Add Search Tab check box, as this will allow your users to search your HTML Help File. Note that if you check this box, you need to check the Compile Full Text search in the Options Dialog (see below). If you wish, you can specify the Navigation pane width too.

Now you need to specify some project options. Click the Project Properties icon . The following dialog will be displayed.




Set the title to "Test HTML Project" , and set the Default Window as Main (the window you just created). If you checked the Add Search Tab option in the Window Properties dialog,



then click Compiler, and select the 'Compile Full Text search information' check box.

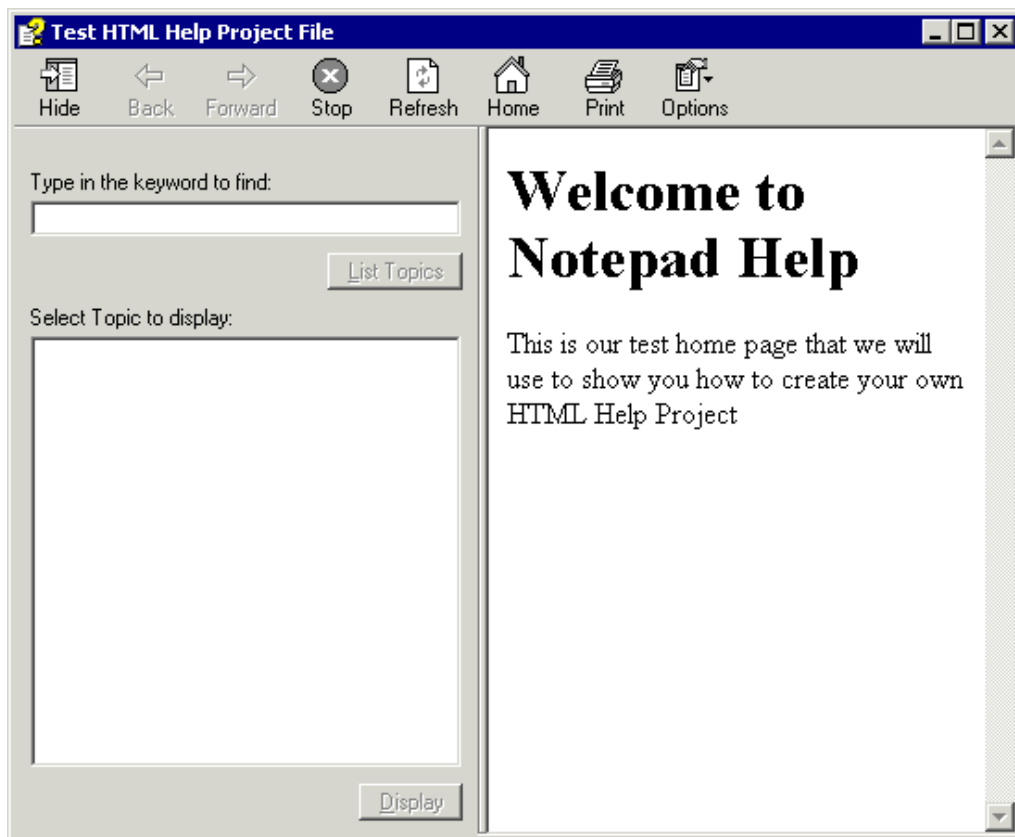
## HTML Files

Now we need to create our first HTML File. Click New from the File Menu, and select HTML File. Click OK, and set the Title to "Test Help Home". Now, you will need to enter some HTML from scratch, or use a tool like FrontPage, and then copy in the HTML. For the moment, just use the code below:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<Title>Test HTMLHelp Home</Title>
</HEAD>
<BODY>
<H1>Welcome to Notepad Help</H1>
<p>This is our test home page that we will use
to show you how to create your own HTML Help Project</p>
</BODY>
</HTML>
```

Now, click File | Save As... and save the file in the **html** directory that we created earlier as home.htm. As this page will be both the default and the home page, go back to the Window Types dialog, and set **Default** and **Home** as html\home.htm. Then, switch back to the Project tab, click the Add/Remove Files  button. Then, click Add, and select the home.htm file you have just saved.

Click the SaveAll button , and then the compile button . Click Compile, and after a few moments your compiled help file will appear (if it does not, check the log that is displayed for errors.). You can also display it by selecting View | Compiled file. You have now successfully completed your first HTML Help file! Your program should look something like this:



Because we have not specified a contents or index file, the Contents and Index tabs are not displayed. We will create these in the next section. As we will need more than one file in order to show you how to use the HTML Help Workshop, create two more html files, using the source below:


**Test Topic 1** (save as Test\_Topic\_1.htm in the [html](#) folder):

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<Title>Test Topic 1</Title>
</HEAD>
<BODY>
<H1>Test Topic 1</H1>
<p>This is Test Topic number 1</p>
</BODY>
</HTML>
```

**Test Topic 2** (save as Test\_Topic\_2.htm in the [html](#) folder):

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<HTML>
<HEAD>
<Title>Test Topic 2</Title>
</HEAD>
<BODY>
<H1>Test Topic 2</H1>
<p>This is Test Topic number 2</p>
</BODY>
```

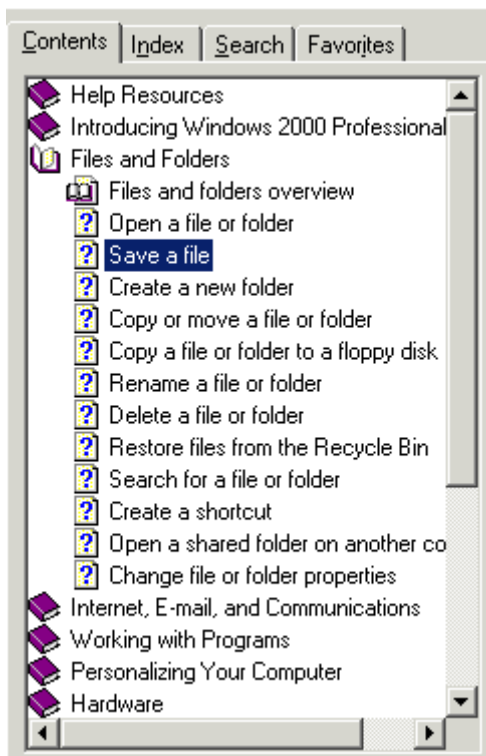
&lt;/HTML&gt;

**Don't forget** to switch back to the Project tab, click the Add/Remove Files  button. Then, click Add, and select the test\_topic\_1.htm and test\_topic\_2.htm files you have just created. You can select them both at the same time by holding down SHIFT, and clicking them.

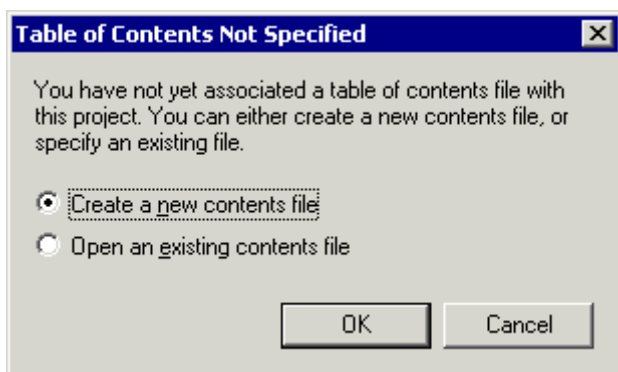
**You will also** need to map [Topics to ID's](#), and use [Style Sheets](#) if you wish.

## Contents Files

The contents files allow you to create a structure so that the user can navigate around your help file. Below is an example.




So, first, click on the Contents tab in your HTML Help Workshop. The following dialog will be displayed:

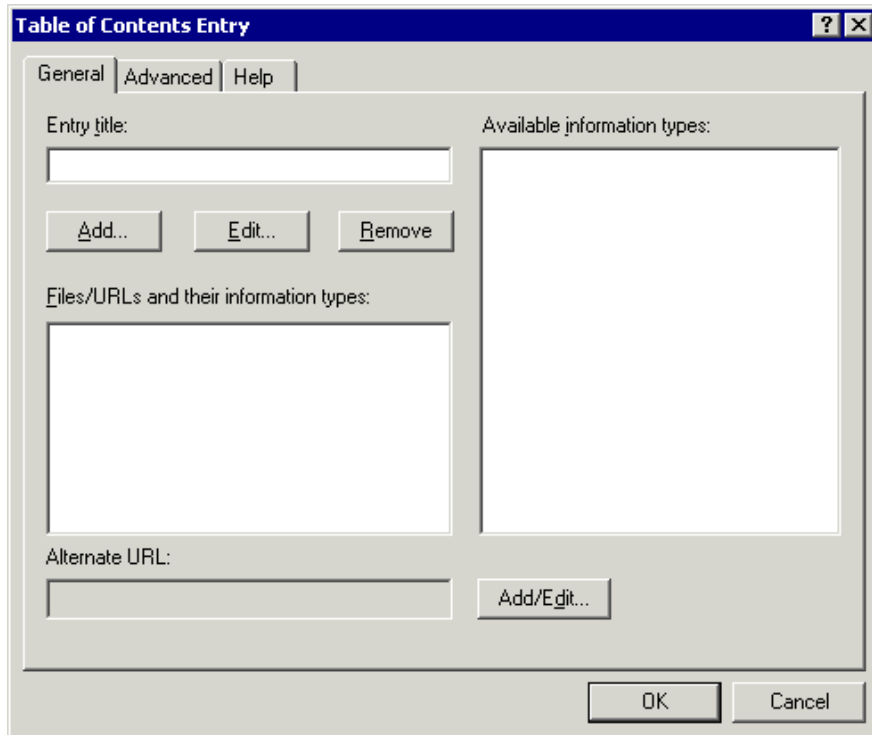



This appears because we have not yet created a contents file. Click OK and one will be created for you. Save it in the **root folder** (**not** in the html folder), using the default name



provided.

If you want to use books (like in the window above), instead of folders, then click the properties button, and un-check the 'Use folders instead of books' option.

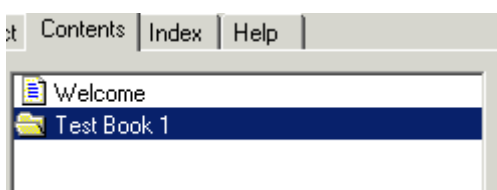
To create a new folder (or book), click the  or the  icon, depending on if the 'Use folders instead of books' option is selected. Then, the window below is displayed:



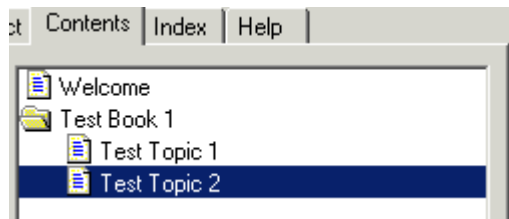
Enter a Title for the folder or book, I have called mine Test Book 1. As we do not want a file to be displayed when the folder is selected, do not add a file to be displayed. If you want to at a later stage, click Add, and specify the file you want to be displayed. Click OK. If you wish, there are some other images you can use for each folder. Click the edit button , and try changing the Image Index property under Advanced.

Now, we need to add some items, or **topics** to the list. First, click the  or the  button (depending on whether you are using folders or books). Click yes to the MsgBox. The same dialog as above will be displayed. Enter Welcome into the Entry Title text box. Then, click Add, so that we can specify a file to be displayed. There are currently no files listed in the next dialog, as we have not yet compiled the project with the contents file included. Click Browse, and select the home.htm file in the [html](#) folder.

Then, click OK on all of the open dialog boxes. You should now see something like this:



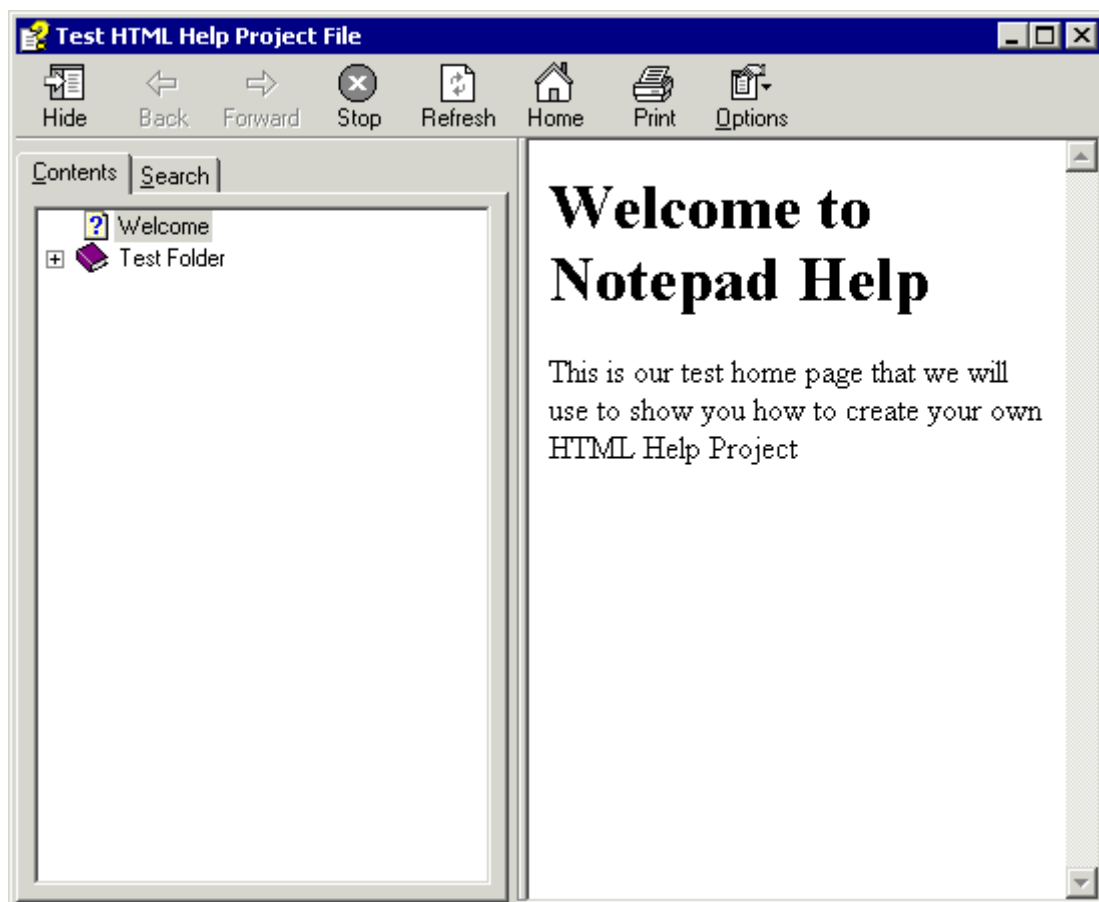
Now, select Test Book 1, and using the Add buttons, add the two other html files to it:



If you need to move folders and/or items around, simply use the arrow buttons:

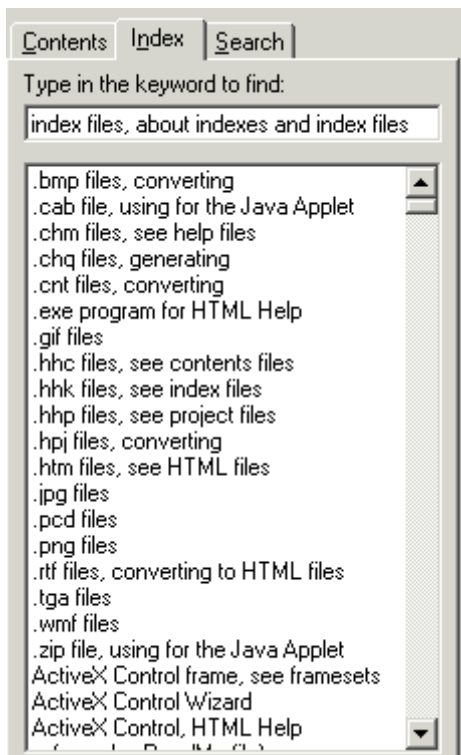



Then, display the Window properties (click on the Project tab, and click the Windows button). Then click the Files tab in the dialog that is displayed, and change **TOC** to Table Of Contents.hhc (or whatever you saved the contents file as). Save all the files, and press compile. Something like this should now be displayed:




## Index Files

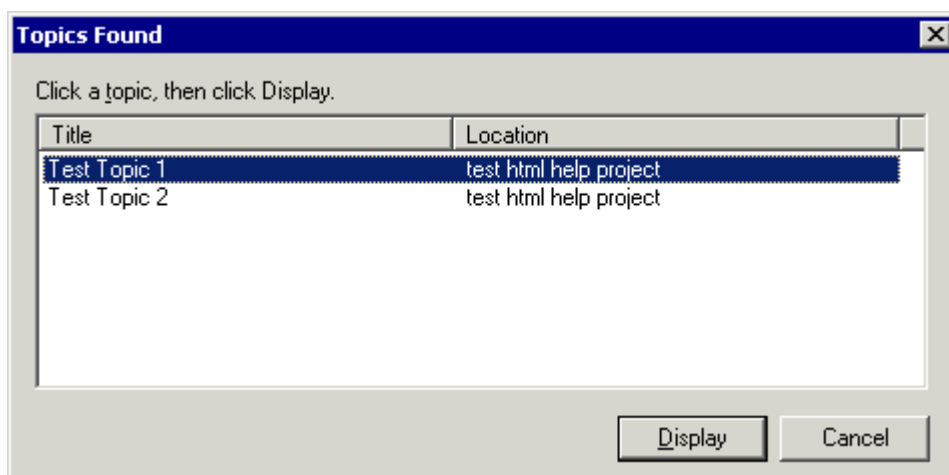
The index files allow you to create a list of keywords, so that when a user clicks a keyword, the appropriate topic is displayed. Here is an example:



First, click New on the File Menu, and select Index tab. A dialog will be displayed asking if you want to create a new one, select that 'Create a new Index file' and click OK. Save the file as Index.hhk in the **root** directory. If you wish, you can click the properties button , and change the font that is used.

To create a keyword, click the new keyword button . Enter the keyword, and then click add to assign the related file to it.


If you add more than one file, then a dialog like this will appear when the user clicks on it:

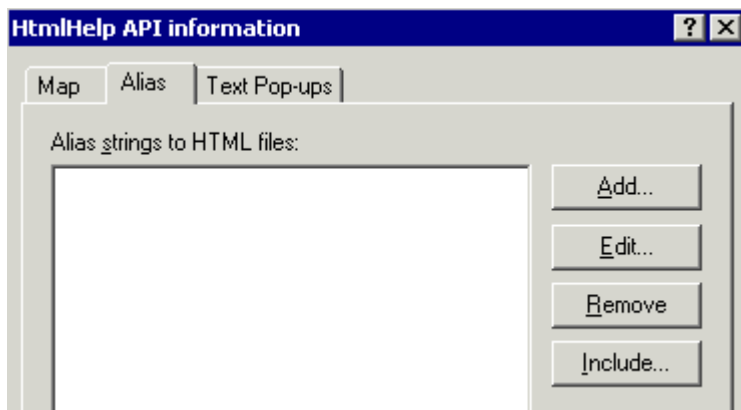


Now, go back to the project tab, and click the Window properties again. Go to the Files tab, and set Index to Index.hhk (or whatever you saved the index file as). Save your project, and compile! You have now got a fully featured HTML Help File. However, obviously, you will need to actually create some true topics, then assign keywords, and create a contents structure.

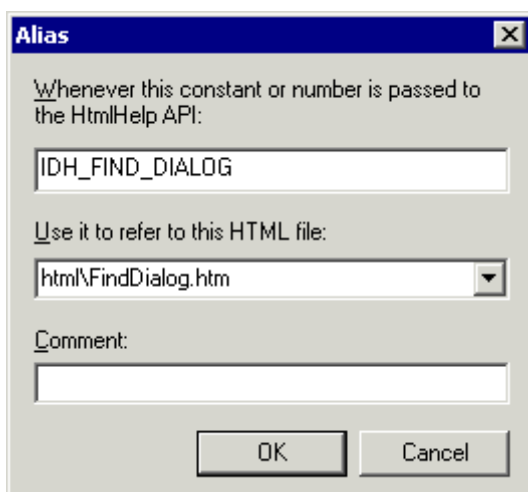
## Mapping Topics to Ids

In order to be able to specify what topic to display from your VB program, you need to map Topics to ID's, and then give the ID's values.

The first thing we need to do is alias the topics to topic ID's. Even if you specify a topic ID within the topic file itself, you may as well do it as a comment, because the workshop has no way to extract that information. Click on the API button , and go to the Alias tab of the resulting dialog box:



Click add to specify a Topic ID for each topic file:



We use the IDH naming system because every one else does - it has become the standard, and who are we to argue? Try to make the Topic ID as clear as possible, so that you can remember what topic it links to, even after a hard days work. Then, select the file you want it to refer to.

Next, we need to assign a value to the Topic ID constant (that we will pass from VB). The HTML Help workshop does not allow you to do this, so, save your work **and close the project**, and go back to notepad. Go to the Alias section, and you will see some ID's, and the files you have assigned them to.

```
[ALIAS]
IDH_FIND_DIALOG=html\find_dialog.htm
IDH_DLG_INSERT_SYMBOLS=html\insert_symbols.htm
IDH_WELCOME=html\welcome.htm
```

Now, go to the Map section (which probably does not actually exist yet, in which case, create it below the Alias section). We need to add some statements that will assign a value to each Topic ID. We use the following syntax:

```
#define TOPIC_ID CONTEXT_INTEGER
```

Here is an example:

```
[MAP]
#define IDH_FIND_DIALOG 1010
#define IDH_DLG_INSERT 1020
#define IDH_WELCOME 1000
```

Again, because every one else does, the Context Integers start from 1000 and work up in increments of 10. Save the file, and reload the file from the HTML Workshop.

## Using Images

If you want to insert images into your HTML Help program (which is very likely!), you need to follow these steps:

- 1) Save the image in the **image** directory.
- 2) Add the file to your project by clicking the Add/Remove Files icon on the Project Tab
- 3) Add the html below in the HTML file where you want to display the image, modifying it to your needs:

```

```

We use the ..\ to tell the compiler to go up a folder, as the HTML file will be in the html directory:

Path	Current Folder
	root\html
..\	root
..\images	root\images

- 4) Save and compile your HTML Help file.

## Using stylesheets

Programmers who already know HTML, and use style sheets can skip this section, as it is exactly the same.

In order to ensure that all your documents are of the same style (ie all the headers are the same size and colour), you can use stylesheets. First, you need to create a stylesheet:

1) Open Notepad, and enter the text below

```
h1 {color:blue;font-size: 12pt;}
p {font-family: 'Tahoma';font-size: 10pt;}
```

2) Save the file as default.css in your **html** directory.

Then, enter the following html into all of your html pages. This will link the html document to the stylesheet. It is therefore a good idea to do this before you have written them all!

```
<link rel="stylesheet" href="default.css" type="text/css">
```

Now, you can have a play around with the stylesheet text. First, lets look at the text we have already entered:

```
; the first bit: 'h1 {' means that the braces {} contain style
information for all text contained in h1 tags.
; the second bit: 'color:blue;' sets the colour of h1 tags to blue
; the third bit: 'font-size: 12pt' sets the font size to 12pts
h1 {color:blue;font-size: 12pt;}
```

So, what's these h1 tags then, I here you ask. Well, instead of surrounding your text by p (p for paragraph) tags, like this:

```
<p>Hello!</p>
```

you use h1 tags, like this:

```
<h1>Hello!</h1>
```

It's as simple as that! In a stylesheet, you can also specify styles for:

Tag	Description
h1	Header 1
h2	Header 2
h3.....h6	and all headers to Header 6
p	Paragraph text (the main body of your text)
a	Hyperlinks

Each

And, inside the braces, you can use the following styles:

Style 'descriptors'	Description
color	Font Color
font-face	Font Name (you need to contain these with a ' each side: font-face:'Tahoma';
font-size	font size. Put a pt after the value: font-size: 10pt;
text-decoration	Set to text-decoration: none; for no underline, and text-decoration: underline; for underlined text
font-weight	Set to font-weight: normal; for regular, and font-weight: bold; for bold text

For more information, click Help Topics on the Help menu, and enter Cascading Style Sheets, and click HTML Tag Reference, and open the CSS book in the contents pane.

Any queries, or if you have a better way to describe this, please [contact me](#).

### Displaying help from a MsgBox

Displaying help from a MsgBox is really quite simple, using the syntax below:

*Msgbox Message, Options + vbMsgBoxHelpButton, DialogTitle, HTMLHelpFilePath & ">" & WindowName, ContextInteger*

So, the following code displays a message box like this



```
MsgBox "Test Error Message Text.", _
    vbOKOnly + vbExclamation + vbMsgBoxHelpButton, _
    "Test Error Message", _
    "C:\TestHTMLHelp.chm" & ">Main", 1010
```

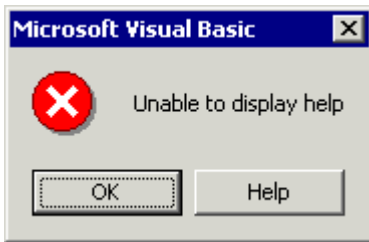
This code presumes the test compiled Help file is saved as C:\TestHTMLHelp.chm, and you have the following items in the Project File:

```
[ALIAS]
IDH_TESTTOPIC2=html\test_topic_2.htm

[MAP]
```

```
#define IDH_FIND_DIALOG 1010
```

If you get this:



check the path, and the Window name. If the HTML Help file is displayed, but with just the default page, check the Context Integer value, and the topic it is assigned to. If you have any problems, please [contact me](#).

## Displaying help

In order to use HTML help in your application, you need to add the following code into a module:

```
' Visual Basic code for implementing HTML Help 1.1

'*****
' Declare the following two constants
' as PUBLIC
Public Const HH_HELP_CONTEXT = &HF           ' display mapped
numeric
Public Const HH_TP_HELP_WM_HELP = &H11       ' text popup help,
same as                                     ' WinHelp HELP_WM_HELP

Private Const HH_DISPLAY_TOPIC = &H0
Private Const HH_HELP_FINDER = &H0           ' WinHelp equivalent
Private Const HH_DISPLAY_TOC = &H1           ' WinHelp equivalent
Private Const HH_DISPLAY_INDEX = &H2         ' WinHelp equivalent
Private Const HH_DISPLAY_SEARCH = &H3        ' not currently
implemented

Private Const HH_SET_WIN_TYPE = &H4
Private Const HH_GET_WIN_TYPE = &H5
Private Const HH_GET_WIN_HANDLE = &H6
Private Const HH_ENUM_INFO_TYPE = &H7       ' Get Info type name,
call                                       ' repeatedly to

enumerate,                               ' -1 at end
                                           ' Add Info type to

Private Const HH_SET_INFO_TYPE = &H8
filter.
Private Const HH_SYNC = &H9
Private Const HH_ADD_NAV_UI = &HA           ' not currently
implemented
Private Const HH_ADD_BUTTON = &HB          ' not currently
implemented
```

```

Private Const HH_GETBROWSER_APP = &HC      ' not currently
implemented
Private Const HH_KEYWORD_LOOKUP = &HD
Private Const HH_DISPLAY_TEXT_POPUP = &HE  ' display string
resource id                                ' or text in a popup
window                                    ' value in dwData
Private Const HH_TP_HELP_CONTEXTMENU = &H10 ' text popup help,
same as                                   ' WinHelp
HELP_CONTEXTMENU
Private Const HH_CLOSE_ALL = &H12          ' close all windows
opened                                   ' directly or
indirectly by                           ' the caller
Private Const HH_ALINK_LOOKUP = &H13       ' ALink version of
                                           ' HH_KEYWORD_LOOKUP
Private Const HH_GET_LAST_ERROR = &H14     ' not currently
implemented
Private Const HH_ENUM_CATEGORY = &H15      ' Get category name,
call                                     ' repeatedly to
enumerate,                              ' -1 at end
Private Const HH_ENUM_CATEGORY_IT = &H16   ' Get category info
type                                    ' members, call
repeatedly to                           ' enumerate, -1 at end
Private Const HH_RESET_IT_FILTER = &H17    ' Clear the info type
filter                                  ' of all info types.
Private Const HH_SET_INCLUSIVE_FILTER = &H18 ' set inclusive
filtering                               ' method for untyped
topics                                  ' to be included in
display                                ' set method for
Private Const HH_SET_EXCLUSIVE_FILTER = &H19 ' topics to be
untyped                                ' the display
excluded from                           ' For Microsoft
Private Const HH_SET_GUID = &H1A           ' dwData is a pointer
Installer --                             ' GUID string
to the                                  ' Used internally.
Private Const HH_INTERNAL = &HFF
' Button IDs

```

```

Private Const IDTB_EXPAND = 200
Private Const IDTB_CONTRACT = 201
Private Const IDTB_STOP = 202
Private Const IDTB_REFRESH = 203
Private Const IDTB_BACK = 204
Private Const IDTB_HOME = 205
Private Const IDTB_SYNC = 206
Private Const IDTB_PRINT = 207
Private Const IDTB_OPTIONS = 208
Private Const IDTB_FORWARD = 209
Private Const IDTB_NOTES = 210 ' not implemented
Private Const IDTB_BROWSE_FWD = 211
Private Const IDTB_BROWSE_BACK = 212
Private Const IDTB_CONTENTS = 213 ' not implemented
Private Const IDTB_INDEX = 214 ' not implemented
Private Const IDTB_SEARCH = 215 ' not implemented
Private Const IDTB_HISTORY = 216 ' not implemented
Private Const IDTB_BOOKMARKS = 217 ' not implemented
Private Const IDTB_JUMP1 = 218
Private Const IDTB_JUMP2 = 219
Private Const IDTB_CUSTOMIZE = 221
Private Const IDTB_ZOOM = 222
Private Const IDTB_TOC_NEXT = 223
Private Const IDTB_TOC_PREV = 224

```

```

Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

```

```

Private Type tagHHN_NOTIFY
    hdr As Variant
    pszUrl As String ' Multi-byte, null-
    terminated string
End Type

```

```

Private Type tagHH_POPUP
    cbStruct As Integer ' sizeof this
    structure
    hinst As Variant ' instance handle for
    string resource
    idString As Variant ' string resource id,
    or text id if pszFile ' is specified in
HtmlHelp call
    pszText As String ' used if idString is
    zero
    pt As Integer ' top center of popup
    window
    clrForeground As ColorConstants ' use -1 for default

```

```

    clrBackground As ColorConstants
    rcMargins As RECT
between edges of window and

member to ignore
    pszFont As String
size, char set, BOLD ITALIC

End Type

Private Type tagHH_AKLINK
    cbStruct As Integer
structure
    fReserved As Boolean
(really!)
    pszKeywords As String
separated keywords
    pszUrl As String
no keywords found (may be

    pszMsgText As String
display in MessageBox if

keyword match
    pszMsgTitle As String
display in MessageBox if

no keyword match
    pszWindow As String
URL in
    fIndexOnFail As Boolean
keyword lookup fails.
End Type

Private Enum NavigationTypes
    HHWIN_NAVTYPE_TOC
    HHWIN_NAVTYPE_INDEX
    HHWIN_NAVTYPE_SEARCH
    HHWIN_NAVTYPE_BOOKMARKS
    HHWIN_NAVTYPE_HISTORY ' not implemented
End Enum

Private Enum IT
    IT_INCLUSIVE
    IT_EXCLUSIVE
    IT_HIDDEN
End Enum

Private Type tagHH_ENUM_IT
    cbStruct As Integer
structure
    iType As Integer

```

```

' use -1 for default
' amount of space

' text, -1 for each

' facename, point

' UNDERLINE

' sizeof this

' must be FALSE

' semi-colon

' URL to jump to if

' NULL)
' Message text to

' pszUrl
' is NULL and no

' Message text to

' pszUrl is NULL and

' Window to display

' Displays index if

' size of this

' the type of the

```

information type i.e.

Exclusive, or Hidden

pszCatName As String  
the Category to

types in a category;

pszITName As String  
the name of the

by call. Caller

freeing

pszITDescription As String  
the description of the

End Type

Private Type tagHH\_ENUM\_CAT

cbStruct As Integer  
structure

pszCatName As String  
the category name

pszCatDescription As String  
the category

End Type

Private Type tagHH\_SET\_INFOTYPE

cbStruct As Integer  
structure

pszCatName As String  
category, if any, the

of.

pszInfoTypeName As String  
info type to add to the

End Type

Private Enum NavTabs

HHWIN\_NAVTAB\_TOP

HHWIN\_NAVTAB\_LEFT

HHWIN\_NAVTAB\_BOTTOM

End Enum

Private Const HH\_MAX\_TABS = 19 ' maximum number of tabs

Private Enum Tabs

HH\_TAB\_CONTENTS

HH\_TAB\_INDEX

HH\_TAB\_SEARCH

HH\_TAB\_BOOKMARKS

' Inclusive,

' Set to the name of

' enumerate the info

' else NULL

' volatile pointer to

' infotype. Allocated

' responsible for

' volatile pointer to

' infotype.

' size of this

' volatile pointer to

' volatile pointer to

' description

' the size of this

' the name of the

' InfoType is a member

' the name of the

' filter

```
HH_TAB_HISTORY
```

```
End Enum
```

```
' HH_DISPLAY_SEARCH Command Related Structures and Constants
```

```
Private Const HH_FTS_DEFAULT_PROXIMITY = (-1)
```

```
Private Type tagHH_FTS_QUERY
```

```
    cbStruct As Integer ' Sizeof structure in
    bytes.
```

```
    fUnicodeStrings As Boolean ' TRUE if all strings
    are unicode.
```

```
    pszSearchQuery As String ' String containing
    the search query.
```

```
    iProximity As Long ' Word proximity.
```

```
    fStemmedSearch As Boolean ' TRUE for
    StemmedSearch only.
```

```
    fTitleOnly As Boolean ' TRUE for Title
    search only.
```

```
    fExecute As Boolean ' TRUE to initiate
    the search.
```

```
    pszWindow As String ' Window to display
    in
```

```
End Type
```

```
' HH_WINTYPE Structure
```

```
Private Const SW_MAXIMIZE = 3
```

```
Private Const SW_MINIMIZE = 6
```

```
Private Const SW_NORMAL = 1
```

```
Private Const SW_SHOW = 5
```

```
Private Type HH_WINTYPE
```

```
    cbStruct As Integer ' IN: size of this
    structure including all
```

```
    fUnicodeStrings As Boolean ' Information Types
    strings are in UNICODE ' IN/OUT: TRUE if all
```

```
    pszType As String ' IN/OUT: Name of a
    type of window
```

```
    fsValidMembers As Variant ' IN: Bit flag of
    valid members
```

```
    fsWinProperties As Variant ' (HHWIN_PARAM_)
    Properties/attributes of the window ' IN/OUT:
```

```
    pszCaption As String ' (HHWIN_)
    title ' IN/OUT: Window
```

```
    dwStyles As Variant ' IN/OUT: Window
    styles
```

```
    dwExStyles As Variant ' IN/OUT: Extended
    Window styles
```

```
    rcWindowPos As RECT ' IN: Starting
```

```
Window styles
```

```
rcWindowPos As RECT
```

position, OUT: current

nShowState As Integer  
(e.g., SW\_SHOW)  
hwndHelp As Variant  
hwndCaller As Variant  
this window

members are only valid if

is set

hwndToolBar As Variant  
in tri-pane window  
hwndNavigation As Variant  
window in tri-pane window  
hwndHTML As Variant  
displaying HTML in tri-pane

iNavWidth As Integer  
navigation window  
rcHTML As RECT  
coordinates

pszToc As String  
table of contents file  
pszIndex As String  
the index file

pszFile As String  
location of the html file  
pszHome As String  
to display when Home

fsToolBarFlags As Variant  
controlling the appearance of the

fNotExpanded As Boolean  
contract or expand, OUT:

curNavType As Integer  
display in the navigational

tabpos As Integer  
HHWIN\_NAVTAB\_TOP, HHWIN\_NAVTAB\_LEFT,  
HHWIN\_NAVTAB\_BOTTOM

idNotify As Integer  
WM\_NOTIFY messages  
tabOrder(HH\_MAX\_TABS + 1) As Byte  
Contents, Index,

Favorites, Reserved 1-5,

cHistory As Integer  
history items to keep

' position  
' IN: show state  
' OUT: window handle  
' OUT: who called  
' The following  
' HHWIN\_PROP\_TRI\_PANE  
' OUT: toolbar window  
' OUT: navigation  
' OUT: window  
' window  
' IN/OUT: width of  
' OUT: HTML window  
' IN: Location of the  
' IN: Location of  
' IN: Default  
' IN/OUT: html file  
' button is clicked  
' IN: flags  
' toolbar  
' IN: TRUE/FALSE to  
' current state  
' IN/OUT: UI to  
' pane  
' IN/OUT:  
' or  
' IN: ID to use for  
' IN/OUT: tab order:  
' Search, History,  
' Custom tabs  
' IN/OUT: number of

```

        pszJump1 As String                                ' (default is 30)
        HHWIN_BUTTON_JUMP1                                ' Text for
        pszJump2 As String                                ' Text for
        HHWIN_BUTTON_JUMP2
        pszUrlJump1 As String                              ' URL for
        HHWIN_BUTTON_JUMP1                                ' URL for
        pszUrlJump2 As String                              ' URL for
        HHWIN_BUTTON_JUMP2
        rcMinSize As RECT                                  ' Minimum size for
        window (ignored in version
        cbInfoTypes As Integer                             ' 1 of the Workshop)
        paInfoTypes;                                       ' size of
    End Type

Private Enum Actions
    HHACT_TAB_CONTENTS
    HHACT_TAB_INDEX
    HHACT_TAB_SEARCH
    HHACT_TAB_HISTORY
    HHACT_TAB_FAVORITES
    HHACT_EXPAND
    HHACT_CONTRACT
    HHACT_BACK
    HHACT_FORWARD
    HHACT_STOP
    HHACT_REFRESH
    HHACT_HOME
    HHACT_SYNC
    HHACT_OPTIONS
    HHACT_PRINT
    HHACT_HIGHLIGHT
    HHACT_CUSTOMIZE
    HHACT_JUMP1
    HHACT_JUMP2
    HHACT_ZOOM
    HHACT_TOC_NEXT
    HHACT_TOC_PREV
    HHACT_NOTES
    HHACT_LAST_ENUM
End Enum

Private Type tagHHNTRACK
    hdr As Variant
    pszCurUrl As String                                    ' Multi-byte, null-
    terminated string
    idAction As Integer                                    ' HHACT_ value
    phhWinType As HH_WINTYPE                              ' Current window type
    structure
End Type

Public Type HH_IDPAIR

```

```

    dwControlId As Long
    dwTopicId As Long
End Type
Public Declare Function htmlhelp Lib "hhctrl.ocx" _
    Alias "HtmlHelpA" (ByVal hwnd As Long, _
    ByVal lpHelpFile As String, _
    ByVal wCommand As Long, _
    ByVal dwData As Long) As Long

```

Then, you need to add a procedure that will set the HTML file path:

```

Public Function SetHTMLHelpStrings() As String
    '// this presumes the help file is in the same directory as
    your app, and Main is the name of the window
    SetHTMLHelpStrings = App.Path & "\TestHTMLHelp.chm" & ">Main"
End Function

```

To display the contents, use this code (from a form, otherwise you will need to change the hwnd value that is passed):

```

Public Sub HTMLHelpContents()
    ' Force the Help window to display
    ' the Contents file (*.hhc) in the left pane
    htmlhelp hwnd, SetHTMLHelpStrings(), HH_DISPLAY_TOC, 0

End Sub

```

To display the index, use this code (from a form, otherwise you will need to change the hwnd value that is passed):

```

Public Sub HTMLHelpIndex()

    ' Force the Help window to display the Index file
    ' (*.hhk) in the left pane
    htmlhelp hwnd, SetHTMLHelpStrings(), HH_DISPLAY_INDEX, 0

End Sub

```

To display a specific topic, using a filename this code (from a form, otherwise you will need to change the hwnd value that is passed):

```

Public Sub HTMLShowTopic(strTopic As String)

    ' Force the Help window to load a specific topic.
    ' The Help window will synchronize the
    ' Contents display automatically
    htmlHelp hwnd, SetHTMLHelpStrings(), HH_DISPLAY_TOPIC, strTopic

End Sub

```

To call it , use this code:

```
HTMLShowTopic "html\test_topic_1.htm"
```

To display a specific topic, using a context id (from a form, otherwise you will need to change the hwnd value that is passed):

```
Public Sub HTMLShowTopic(lngTopicID As Long)

    ' Force the Help window to load a specific topic.
    ' The Help window will synchronize the
    ' Contents display automatically
    htmlHelp hwnd, SetHTMLHelpStrings(), HH_HELP_CONTEXT, lngTopicID

End Sub
```

To call it , use this code:

```
HTMLShowTopic 1000 '// 1000 = Context ID
```

At the moment, you cannot display the Search tab automatically from VB