

HTML Help API overview

The HTML Help application programming interface (API) enables a Windows program to create a help window that displays a help topic. The Windows program has complete control over the type, style, and position of the help window.

The fundamental feature of the HTML Help API is the [help window](#).

Through the API commands, you can create a help window that hosts a Microsoft Internet Explorer DLL (Shdocvw.dll) and displays an HTML file that you specify. The help window is owned by the window you specify. As an owned window, a help window automatically stays on top of its owner and closes when the owner is closed.

You define the styles, coordinates, captions, and display state of the help window.

- Context-sensitive help.
 - Keyword lookup.
 - Interaction between a Windows program and a compiled help (.chm) file.
- In addition to creating help windows, the HTML Help API commands also enable you to provide:
- Control over the Navigation pane in the standard HTML Help Viewer.

ANSI and Unicode versions of the HTML Help

API When a program is compiled ANSI, the ANSI version of the API is used by default, allowing you to specify either HtmlHelpA() or HtmlHelp().

- The HTML Help API function is modeled after the WinHelp API function to simplify the process of updating existing programs from WinHelp to HTML Help. However, the HTML Help API command names and parameters are not the same as those in the WinHelp API, and the output file formats are very different.

Note

The HTML Help API has one function that displays a help window. Using the API commands, you can specify which topic to display in the help window, whether the help window is a three-pane Help Viewer or a pop-up window, and whether the HTML topic file should be accessed via a context ID, an [HTML Help URL](#), or a Keyword link (KLink) lookup.

HTML Help API syntax

HWND HtmlHelp(

 HWND *hwndCaller*,

 LPCSTR *pszFile*,

 UINT *uCommand*,

 DWORD *dwData*);

Parameter	Description
<i>hwndCaller</i>	<p>Specifies the handle (hwnd) of the window calling HtmlHelp(). The help window is owned by this window.</p> <p>When the help window is closed, HtmlHelp() will return focus to the owner unless the owner is the desktop. If <i>hwndCaller</i> is the desktop, then the operating system determines where focus is returned.</p> <p>In addition, if HtmlHelp() sends any notification messages from the help window, they are sent to <i>hwndCaller</i> as long as you have enabled notification message tracking in the help window definition.</p>
<i>pszFile</i>	<p>Depending on the <i>uCommand</i> value, specifies the file path to either a compiled help (.chm) file, or a topic file within a specified help file.</p> <p>A window type name can also be specified, preceded with a greater-than (>) character.</p> <p>If the specified command does not require a file, this value may be NULL.</p>

<i>uCommand</i>	Specifies the command to complete.
<i>dwData</i>	Specifies any data that may be required, based on the value of the <i>uCommand</i> parameter.

- The handle (hwnd) of the help window.

Return value

NULL. In some cases, NULL indicates failure; in other cases, NULL indicates that the help window has not yet been created.

Depending on the specified *uCommand* and the result, HtmlHelp() returns one or both of the following:

Example

The following example calls the [HH_DISPLAY_TOPIC](#) command to open the help file named Help.chm and display its default topic in the help window named Mainwin. Generally, the help window specified in this command is a standard [HTML Help Viewer](#).

```
HWND hwnd =
```

```
    HtmlHelp(  
        GetDesktopWindow(),  
        "c:\\Help.chm::/Intro.htm>Mainwin",  
        HH_DISPLAY_TOPIC,  
        NULL);
```

Comments

When using the HTML Help API, set the stack size of the hosting executable to at least 100k. If the defined stack size is too small, then the thread created to run HTML Help will also be created with this stack size, and failure could result. Optionally, you can remove `/STACK` from the link command line, and remove any `STACK` setting in the executable's DEF file (default stack size is 1MB in this case). You can also you can set the stack size using the `/Fnumber` compiler command (the compiler will pass this to the linker as `/STACK`).

 [About the HTML Help API](#)

Accessing the HTML Help API

The functionality provided by the HTML Help API resides in the HTML Help ActiveX control (Hhctrl.ocx), which is installed when you set up HTML Help Workshop.

To gain access to the HTML Help API, you link to the Htmlhelp.lib file and include the Htmlhelp.h file in your Windows program. Both of these files are installed on your system when you set up HTML Help Workshop.

About Htmlhelp.lib

Htmlhelp.lib is an export library that exposes the HTML Help API and loads Hhctrl.ocx only when HtmlHelp() is called. In addition, Htmlhelp.lib locates the registered Hhctrl.ocx.

By default, Htmlhelp.lib is located in the following directory:

C:\Program Files\HTML Help Workshop\Lib

About Htmlhelp.h

Htmlhelp.h is a header file that contains the declarations for HtmlHelp(). It must be included in your Windows program.

By default, Htmlhelp.h is located in the following directory:
C:\Program Files\HTML Help Workshop\Include

- If you are using the HTML Help API with Windows 95 and Internet Explorer 3.x, you must either set up [DCOM for Microsoft Windows® 95, version 1.2](#), or set up [Internet Explorer 4.x](#). Otherwise, the calls that you make to HtmlHelp() may not be processed correctly.

Notes

 [About the HTML Help API](#)

About HTML Help URLs

Many of the HTML Help API commands require that you specify an *HTML Help URL* in the *pszURL* parameter.

An HTML Help URL specifies a compiled help (.chm) file or a topic within a help file. Usually, a [window type](#) in which to display the HTML Help URL is also specified.

Helpfile.chm::/Topic.htm[>Window name]

Specifying an HTML Help URL
The following example shows the syntax to display a compiled help file:
Helpfile.chm::/Topic.htm[>Window name]
where Helpfile.chm is the name of the compiled help file, Topic.htm is the name of the HTML file that you want to open and Window name is the name of the help window in which you want the topic to appear.
The following example shows the syntax to display a compiled help file:
To specify a topic within a compiled help file, use the following syntax:


Specifying a topic file path

A compiled help file retains the folder structure in which it was organized before compilation, unless the option to compile flat has been selected.

For example, if a project is organized in three folders (one for HTML files, one for images, and one for style sheets), the help file will contain those same folders internally. The folder in which the file resides is considered the root folder in the HTML Help ActiveX control, you must where Overview is a folder within the Html folder.

To correctly link to a topic file, you must specify the full path. The prefix ms-its, which determines where the help file resides on a user's computer. The prefix ms-its is a pluggable protocol that follows standards set up by the [World Wide Web Consortium \(W3C\)](#).

The ms-its protocol works with Microsoft Internet Explorer 4.0 or later, but is not supported in all browsers.

- `ms-its:Helpfile.chm::/Topic.htm[>Window name]`
- The prefix `mk:@MSITStore` is an earlier version of the ms-its protocol that works with Microsoft Internet Explorer 3.0 or later.
`mk:@MSITStore:helpfile.chm::/Topic.htm[>Window name]`
- The following examples show the syntax to use when including the prefix:
 [About the HTML Help API](#)

Window types

A window type is used to display a help topic and to provide users with a way to navigate through a compiled help (.chm) file. A help window is based on an [HH_WINTYPE](#) structure and hosts a Microsoft Internet Explorer DLL (Shdocvw.dll) to display a help topic.

Any help window that you create through the HTML Help API is owned by the calling, or parent, program. This allows the help window to stay on top of its parent, yet not be on top of any other program that has focus.

In addition, any help window that you create through the API is bound by the actions of its owner. For example, if the user minimizes the calling program, the help window goes away until the owner is restored. If the user exits the calling program, the help window also closes without requiring any action from its owner. Because the calling program created the help window, it can change the position and properties of that window without affecting the procedural help of any other program.

Defining window types

Either the help author can define window types in the project (.hhp) file using HTML Help Workshop, or the developer can define window types programmatically using the HTML Help API.

A window type definition, such as the standard [HTML Help Viewer](#), includes all of the necessary information about the window such as window style attributes and any navigational elements (such as a table of contents, index, or full-text search). When the window type is created, part of its definition are the file locations of its navigational elements. These navigational elements do not change during the life of the window type.

Window types are specific to the compiled help (.chm) file for which they are created, unless the window type is a [global window](#). Associating a window type with a single compiled help file enables you to create multiple help files that all contain a window type of the same name. For example, you could create fifty compiled help files and define a window type named Main in each file without the different window types conflicting with each other.

The total number of window types that can be defined, and the total number of windows that can be displayed at any one time, is limited solely by available memory and system resources.

Opening a help window

To create or access a help window, the developer can call the HTML Help API or the user can double-click a compiled help (.chm) file.

If a call to HtmlHelp() is made specifying a window type that does not exist, a type will be created using the specified name, but with default properties.

When a user double-clicks a compiled help (.chm) file, the HTML Help executable program (Hh.exe) calls the HTML Help API and the correct help window and topic appears. Accessing a help window using this method is slightly different than invoking a window through the HTML Help API.

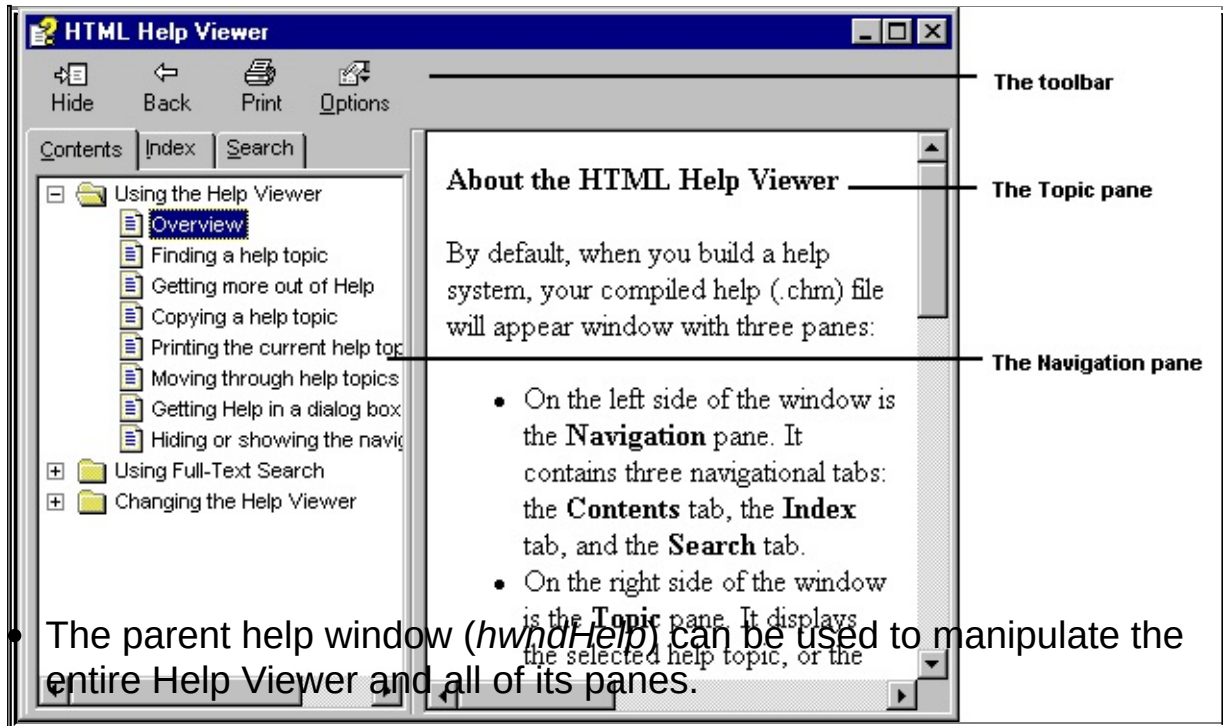
By default, when a user double-clicks a help (.chm) file, the file opens in the window type defined in the help project (.hhp) file and displays the default help topic that is also defined in the project file. If no window is defined in the project file, then the default [Help Viewer](#) is used. If no default topic is defined in the project file, then a random help topic appears.

 [About the HTML Help API](#)

About the HTML Help Viewer

The HTML Help Viewer is the standard three-pane help window in which a compiled help (.chm) file appears. Each pane of the Help Viewer can be manipulated through an [HH_WINTYPE](#) structure.

The Help Viewer is shown below. Click a pane for more information:



 [About window types](#)
Note

About global window types

A global window is a help window that can be used to display one or more compiled help (.chm) files. Global window types provide backward compatibility for a bug in HTML Help version 1.1/1.1a. Unlike HTML Help v1.3, previous versions did not require a window type to be owned by a specific compiled help (.chm) file.

By default, all [window types](#) are specific to the compiled help (.chm) file for which they were defined. That is, a window type is owned by a particular .chm file. However, after you define a global window type, it is not specific to a particular .chm file and can be shared by an unlimited number of .chm files.

A popular use for global window types is to provide a simple help window to display a topic from any .chm file. Global window types are most useful when they do not include a Navigation pane because the data in the Navigation pane does not update when a different .chm is loaded.

- Call the [WM_SET_WINDOW_TYPE](#) command with NULL as the value of the *pszFile* parameter.

Defining global window types

- Add the prefix \$global_ to your window type name.

There are two ways to define a global window type:

For example, MyWindow is a standard window type name, but \$global_MyWindow is a global window type name.

Working with global windows

Like a standard window type, which is associated with a single set of navigational elements (toc, index, and full text search), a global window type is also associated with navigational elements that cannot be modified. Therefore, even though you can use one global window type to view multiple .chm files, the global window will not update the table of contents (.hhc) file, the index (.hhk) file, or any other navigational element to match the current .chm file.

For example, if A.chm and B.chm each have a global window type named \$global_main, the two definitions will conflict with each other. In most cases, the first global window type that is created controls the navigational elements that display. This means that if A.chm creates \$global_main first, you will see the navigational elements defined in A.chm. However, if B.chm creates \$global_main first, you will see the navigational elements defined in B.chm. After a global window is created,

- If you plan to use global windows, define only one global window type for a single .chm file and make sure to load that .chm before loading any other .chm that uses the global window type.

Notes After you define a global window type, all navigational elements are frozen. Regardless of the .chm file being displayed, the table of contents, index, and search information is based on the first .chm that is loaded in the window.

- Due to a bug in HTML Help v1.1/1.1a, all window types were treated as global. HTML Help v1.3 provides global window types to ensure backward compatibility.

 [About window types](#)

Pop-up windows. The base class for all POPUP structures specifies the following default properties:

- If there is insufficient screen real estate to display the text, the text is clipped. A pop-up window never includes a scroll bar.
- The pop-up window is dismissed when the user clicks inside or outside the window, presses Esc, or when the pop-up window loses focus.

 [About window types](#)

About the Reference Section

examples for all commands (*uCommand* parameter values) that can be passed to `HtmlHelp()`.

The reference section provides comprehensive information about how to use the HTML Help API commands, data structures, and notification messages. There are three sections:

- The [structures reference](#) provides syntax descriptions for all data structures used by `HtmlHelp()`.
- The [notification messages reference](#) provides syntax descriptions for all notification messages sent by `HtmlHelp()`.

About commands

The HTML Help API supports a set of commands (*uCommand* values) that enable you to access functionality programmatically.

The most basic of the commands is [HH_DISPLAY_TOPIC](#), which enables you to open a compiled help (.chm) file in a specific help window and display a specific topic within the help file.

Commands quick reference

- [HH_CLOSE_ALL](#)
- [HH_GET_WIN_HANDLE](#)

Category	Commands
Window types	<ul style="list-style-type: none"> • HH_DISPLAY_TEXT_POPUP • HH_DISPLAY_TOPIC • HH_SET_WIN_TYPE • HH_HELP_CONTEXT
Context-sensitive help	<ul style="list-style-type: none"> • HH_API_HELP_CONTEXTMENU • HH_KEYWORD_LOOKUP
Keyword lookups	<ul style="list-style-type: none"> • HH_DISPLAY_INDEX • HH_DISPLAY_SEARCH
Navigation pane	<ul style="list-style-type: none"> • HH_DISPLAY_TOC • HH_GET_LAST_ERROR
Error messages	<ul style="list-style-type: none"> • HH_SYNC
Contents synchronization	<ul style="list-style-type: none"> • HH_INITIALIZE • HH_PRETRANSLATEMESSAGE • HH_UNINITIALIZE
Single threading	

 [About the HTML Help API](#)

HH_ALINK_LOOKUP command

Looks up one or more Associative link (ALink) names in a compiled help (.chm) file.

The ALink names to search for, and the action to be taken if no matches are found, are specified in the **HH_AKLINK** structure.

<i>pszFile</i>	<i>dwData</i>
Specifies the compiled help (.chm) file that contains ALink names.	Points to an HH_AKLINK structure.

Example

```
HH_AKLINK link;

link.cbStruct = sizeof(HH_AKLINK);

link.fReserved = FALSE;

link.pszKeywords = "open";

link.pszUrl = NULL;

link.pszMsgText = NULL;

link.pszMsgTitle = NULL;

link.pszWindow = NULL;

link.fIndexOnFail = TRUE;
```

```
HtmlHelp(

    GetDesktopWindow(),

    "c:\\MyHelp.chm",

    HH_ALINK_LOOKUP,

    (DWORD) &link);
```

Return value

The handle (hwnd) of the help window.

- Help authors insert ALink names into target topic files using the HTML **Comments** **Information** feature.
- An ALink name lookup can also be invoked using the HTML Help ActiveX control **ALink** command.
- ALink name/KLink keyword lookups are case sensitive. Multiple keywords are delimited by a semicolon.

See also

[HH_KEYWORD_LOOKUP](#)

 [About commands](#)

HH_CLOSE_ALL command

Closes all windows opened directly or indirectly by the calling program.

<i>hwndCaller</i>	<i>pszFile</i>	<i>dwData</i>
Must be NULL.	Must be NULL.	Must be zero.

Example

```
HtmlHelp(  
    NULL,  
    NULL,  
    HH_CLOSE_ALL,  
    0);
```


Return value

NULL.

the calling program.

Comments

 [About commands](#)

HH_DISPLAY_INDEX command

Selects the **Index** tab in the Navigation pane of the HTML Help Viewer and searches for the keyword specified in the *dwData* parameter.

<i>pszFile</i>	<i>dwData</i>
Specifies a compiled help (.chm) file, or a specific topic within a compiled help file.	Specifies the keyword to select in the index (.hhk) file.

Example

```
HtmlHelp(  
    hwnd,  
    "cat.chm",  
    HH_DISPLAY_INDEX,  
    (DWORD)"meow");
```

Return value

The handle (hwnd) of the help window.

See also

[HH_DISPLAY_TOC](#)

[HH_DISPLAY_SEARCH](#)

 [About commands](#)

HH_DISPLAY_SEARCH command

Selects the **Search** tab in the Navigation pane of the HTML Help Viewer, but does not actually perform a search.

<i>pszFile</i>	<i>dwData</i>
Specifies a compiled help (.chm) file, or a specific topic within a compiled help file.	Specifies a pointer to an HH_FTS_QUERY structure.

Example

```
HH_FTS_QUERY q ;
```

```
HtmlHelp(  
    hwnd,  
    "cat.chm",  
    HH_DISPLAY_SEARCH,  
    (DWORD)&q);
```


Return value

The handle (hwnd) of the help window.

See also

[HH_DISPLAY_INDEX](#)

[HH_DISPLAY_TOC](#)

 [About commands](#)

- A text string based on a resource ID.

HH_DISPLAY_TEXT_POPUP command

Opens a string ID based dialog that displays the contents of a compiled help (.chm) file.

following:

<i>pszFile</i>	<i>dwData</i>
To use an explicit text string: Specify a NULL value.	
To use a text string from a resource: Specify a NULL value.	Specifies a pointer to an HH_POPUP structure.
To use text string from a text file contained in a compiled help file: Specify the .chm file and the text file within the .chm file.	

Example

```
HtmlHelp(  
    hwndCTRL,  
    NULL,  
    HH_DISPLAY_TEXT_POPUP,  
    (DWORD)&popup);
```

Return value

The handle (hwnd) of the pop-up window.

See also

[HH_TP_HELP_CONTEXTMENU](#)

[HH_TP_HELP_WM_HELP](#)

 [About commands](#)

HH_DISPLAY_TOC command

Selects the **Contents** tab in the Navigation pane of the HTML Help Viewer.

<i>pszFile</i>	<i>dwData</i>
Specifies a compiled help (.chm) file, or a specific topic within a compiled help file.	Specifies NULL or a pointer to a topic within a compiled help file.

Example

```
HtmlHelp(  
    GetDesktopWindow(),  
    "c:\\MyHelpFile.chm::/intro.htm",  
    HH_DISPLAY_TOC,  
    NULL);
```


Return value

The handle (hwnd) of the help window.

Comments

See also

[HH_DISPLAY_INDEX](#)

[HH_DISPLAY_SEARCH](#)

[HH_DISPLAY_TOPIC](#)

 [About commands](#)

HH_DISPLAY_TOPIC command

Opens a help topic in a specified help window.

If a window type is not specified, a default window type is used. If the window type or default window type is open, the help topic replaces the current topic in the window.

<i>pszFile</i>	<i>dwData</i>
Specifies a compiled help (.chm) file, or a specific topic within a compiled help file. To specify a defined window type, insert a greater-than (>) character followed by the name of the window type.	Specifies NULL or a pointer to a topic within a compiled help file.

Example

```
HWND hwnd =
```

```
    HtmlHelp(
```

```
        GetDesktopWindow(),
```

```
        "c:\\help.chm::/intro.htm>mainwin",
```

```
        HH_DISPLAY_TOPIC,
```

```
        NULL);
```

Return value

The handle (hwnd) of the help window.

- A default help window contains only the Topic pane and is not a three-pane window.

Comments

See also

[HH_HELP_CONTEXT](#)

 [About commands](#)

HH_GET_LAST_ERROR command

Returns information about the last error that occurred in the HTML Help ActiveX control (Hhctrl.ocx).

<i>pszFile</i>	<i>dwData</i>
Must be NULL.	A pointer to a HH_LAST_ERROR structure.

Example

```
USES_CONVERSIONS; // For Unicode to ANSI string conversion
```

```
HH_LAST_ERROR lasterror ;
```

```
HWND hwnd = HtmlHelp(  
    hOwner,  
    NULL,  
    HH_GET_LAST_ERROR,  
    reinterpret_cast<DWORD>(&lasterror;));
```

```
// Make sure that HH_GET_LAST_ERROR succeeded.
```

```
if (hwnd != 0)
```

```
{
```

```
    // Only report an error if we found one:
```

```
    if (FAILED(lasterror.hr))
```

```
    {
```

```
        // Is there a text message to display...
```

```
if (lasterror.description)
{
    // Convert the String to ANSI
    TCHAR* pDesc = OLE2T(lasterror.description);
    ::SysFreeString(lasterror.description);

    // Display
    MessageBox(hOwner, pDesc,
    "Help Error", MB_OK);
}
}
}
```

Return value

a message box or a text file, and/or responding to an error.

Comments This function reports on a limited range of possible errors. If a call to HtmlHelp() returns NULL, **HH_GET_LAST_ERROR** may not always report an error.

 [About commands](#)

HH_GET_WIN_HANDLE command

Returns the handle (hwnd) of a specified window type.

<i>pszFile</i>	<i>dwData</i>
Specifies the name of the compiled help (.chm) file in which the window type is defined.	Specifies the name of the window type whose handle you want to return.

Example

```
HtmlHelp(  
    hwndCaller,  
    "c:\\MyHelpFile.chm,  
    HH_GET_WIN_HANDLE,  
    (DWORD) "MyWindowType") ;
```

- NULL, if the help window has not yet been created.

Return value

type is not defined in a compiled help file, then specify NULL.

Comments

See also

[HH_GET_WIN_TYPE](#)

[HH_SET_WIN_TYPE](#)

 [About commands](#)

HH_GET_WIN_TYPE command

Retrieves a pointer to the **HH_WINTYPE** structure associated with a specified window type.

<i>pszFile</i>	<i>dwData</i>
<p>Specifies the name of the window type whose information you want to get and the name of the compiled help (.chm) file in which the window type is defined.</p> <p>The window name must begin with a greater-than (>) character and must be preceded by the name of the compiled help file it is defined in.</p>	<p>Specifies the address of a pointer to an HH_WINTYPE structure.</p> <p>Deep copy the structure to which <i>dwData</i> points before modifying the structure.</p>

Example

```
HH_WINTYPE* pWinType ;  
  
HtmlHelp(  
    GetDesktopWindow(),  
    "Help.chm>mainwin",  
    HH_GET_WIN_TYPE,  
    (DWORD) &pWinType);
```

- On success, NULL, if the help window has not yet been created.

Return value

- On failure, -1, if the specified window type has not been defined.

defined in a .chm file, then specify NULL.

Comments

See also

[HH_SET_WIN_TYPE](#)

 [About commands](#)

HH_HELP_CONTEXT command

Displays a help topic based on a mapped topic ID.

If a window type is not specified, a default window type is used. If the window type or default window type is open, the help topic replaces the current topic in the window.

<i>pszFile</i>	<i>dwData</i>
Specifies the compiled help (.chm) file that contains the mapping information. To specify a defined window type, insert a greater-than (>) character followed by the name of the window type.	Specifies the numeric ID of the topic to display. You must map symbolic IDs of dialog boxes to numeric IDs in the [MAP] section of your project (.hhp) file.

Example

```
HtmlHelp(  
    GetDesktopWindow(),  
    "help.chm",  
    HH_HELP_CONTEXT,  
    5000);
```

Comments

See also

[HH_DISPLAY_TOPIC](#)

 [About commands](#)

HH_INITIALIZE command

This command initializes the help system for use and must be the first HTML Help command called. It returns a cookie which must be used in the [HH_UNINITIALIZE](#) call.

HH_INITIALIZE configures HTML Help to run on the same thread as the calling application instead of a secondary thread by setting the global property **HH_GPROPID_SINGLETHREAD** to `VARIANT_TRUE`. Running HTML Help on the same thread as the calling application requires the calling application to send messages to HTML Help by calling the [HH_PRETRANSLATEMESSAGE](#) command.

<i>pszFile</i>	<i>dwData</i>
Must be NULL.	Specifies a pointer to a DWORD. This call returns a cookie that you must pass as the value of <i>dwData</i> when you call HH_UNINITIALIZE .

Example

```
DWORD dwCookie = NULL;
```

```
HtmlHelp(  
    NULL,  
    NULL,  
    HH_INITIALIZE,  
    (DWORD)&dwCookie); // Cookie returned by Hhctrl.ocx.
```

these commands once during the life of your application.

Comments

See also

[HH_PRETRANSLATEMESSAGE](#)

[HH_UNINITIALIZE](#)

 [About commands](#)

HH_KEYWORD_LOOKUP command

Looks up one or more keywords in a compiled help (.chm) file.

The keywords to search for and the action to be taken if no matches are found are specified in the **HH_AKLINK** structure.

<i>pszFile</i>	<i>dwData</i>
Specifies the compiled help (.chm) file that contains keywords.	Points to an HH_AKLINK structure.

Example

```
HH_AKLINK link;  
  
link.cbStruct = sizeof(HH_AKLINK) ;  
  
link.fReserved = FALSE ;  
  
link.pszKeywords = "open" ;  
  
link.pszUrl = NULL ;  
  
link.pszMsgText = NULL ;  
  
link.pszMsgTitle = NULL ;  
  
link.pszWindow = NULL ;  
  
link.fIndexOnFail = TRUE ;
```

```
HtmlHelp(  
    GetDesktopWindow(),  
    "c:\\myhelp.chm",  
    HH_KEYWORD_LOOKUP,  
    (DWORD)&link);
```

Return Value

The handle (hwnd) of the help window.

- Help authors insert keywords into topic files using the HTML Help **Comments** **Information** feature.
- A keyword lookup can also be invoked using the HTML Help ActiveX control **KLink** command.
- ALink Name/Keyword lookups are case sensitive, and multiple keywords are delimited by a semicolon (;).

See Also

[HH_ALINK_LOOKUP](#)

 [About commands](#)

HH_PRETRANSLATEMESSAGE command

This command is called in the message loop of your Windows application to ensure proper handling of Windows messages, especially keyboard messages when running HTML Help single thread.

The HTML Help API is *not* thread safe and must be called from one and only one thread in a process.

<i>pszFile</i>	<i>dwData</i>
Must be NULL.	Points to a Win32 MSG structure.

Example

```
MSG msg;
```

```
while (GetMessage (&msg, NULL, 0, 0)) //Retrieve a message from the
```

```
{ //calling thread's message queue
```

```
if (!HtmlHelp (
```

```
    NULL,
```

```
    NULL,
```

```
    HH_PRETRANSLATEMESSAGE,
```

```
    &msg;))
```

```
{
```

```
    TranslateMessage (&msg);
```

```
    DispatchMessage (&msg);
```

```
}
```

```
}
```

Return value

- Before calling this command, you must first set the global property `COMPONENTS_SINGLETHREAD` to `VARIANT_TRUE` by calling the [HH_INITIALIZE](#) command.

See also

[HH_INITIALIZE](#)

[HH_UNINITIALIZE](#)

 [About commands](#)

HH_SAFE_DISPLAY_TOPIC command

Opens a help topic in a specified help window, and disables all shortcuts in the current process. Shortcuts will remain disabled as long as the calling process is active. This command can be used in place of HH_DISPLAY_TOPIC.

If a window type is not specified, a default window type is used. If the window type or default window type is open, the help topic replaces the current topic in the window.

<i>pszFile</i>	<i>dwData</i>
Specifies a compiled help (.chm) file, or a specific topic within a compiled help file. To specify a defined window type, insert a greater-than (>) character followed by the name of the window type.	Specifies NULL or a pointer to a topic within a compiled help file.

Example

HWND hwnd =

HtmlHelp(

 GetDesktopWindow(),

 "file:///c:\\help.chm::/intro.htm>mainwin",

 HH_SAFE_DISPLAY_TOPIC,

 NULL);

Return value

The handle (hwnd) of the help window.

Comments

A default help window contains only the Topic pane and is not a three-pane Help Viewer.

- http:

The `https://SAFE_DISPLAY_TOPIC` command will only work with the following standard protocols:

- ms-its.
- its:
- file:
- mk@msitstore:

See also

[HH_HELP_CONTEXT](#)

[HH_DISPLAY_TOPIC](#)

 [About commands](#)

HH_SET_WIN_TYPE command

Creates a new help window or modifies an existing help window at run time.

<i>pszFile</i>	<i>dwData</i>
<p>Specifies the name of the window type that you want to create or modify and the name of the compiled help (.chm) file in which the window type is defined.</p> <p>The window type name must begin with a greater-than (>) character and must be preceded by the name of the compiled help file in which it is defined.</p>	<p>Points to an HH_WINTYPE structure.</p>

Example

```
HH_WINTYPE WinType ;
```

```
...
```

```
HtmlHelp(  
    GetDesktopWindow(),  
    "..\\help.chm>mainwin",  
    HH_SET_WIN_TYPE,  
    (DWORD) &WinType);
```


On failure, NULL, if the specified window type has not been defined.

Return value

- Always specify the name of the compiled help file in which the window type is defined when calling an API command. If the window type is not defined in a help file, then specify NULL.
- If a NULL is passed to **HH_SET_WIN_TYPE**, the name of the window type is placed into an array and is treated as a [global window type](#). A global window type is a window type that is not specific to a particular compiled help file. Only use global window types if you require backward compatibility for existing applications.

See also

[HH_GET_WIN_TYPE](#)

 [About commands](#)

HH_SYNC command

Locates and selects the contents entry for the help topic that is open in the Topic pane of the HTML Help Viewer.

<i>pszFile</i>	<i>dwData</i>
<p>Specifies the name of the window type that you want to sync and the name of the compiled help (.chm) file in which the window type is defined.</p> <p>The window type name must begin with a greater-than (>) character and must be preceded by the name of the compiled help file in which it is defined.</p>	<p>Specifies a pointer to a topic within a compiled help file. This value is the topic file to which the contents will synchronize.</p>

Example

```
HtmlHelp(  
    hwndCaller,  
    "..\\MyHelpFile.chm>wintype",  
    HH_SYNC,  
    "MyHelpFile.chm::/html\\MyTopic.htm");
```

toolbar of the Help Viewer.

Comments

 [About commands](#)

HH_TP_HELP_CONTEXTMENU command

Opens a pop-up context menu. Generally used in response to the Windows **WM_CONTEXTMENU** message. For example, this message is sent when a user right-clicks a dialog box control.

<i>hwndCaller</i>	<i>pszFile</i>	<i>dwData</i>
Specifies the window handle of the dialog box control for which you want pop-up help to appear. This is typically the control that has focus.	<p>Specifies the compiled help (.chm) file, and the text file that contains the pop-up help topics.</p> <p>By default, the text file is named Cshelp.txt. If Cshelp.txt is located in the root of the compiled help file, then you only need to specify the help file name. If not, you must also specify the relative path.</p>	<p>Specifies an array of DWORDs containing pairs of dialog box control IDs and help topic IDs. The array must be terminated by zero. If a control does not require pop-up help, specify -1 for the help topic ID.</p> <pre>DWORD ids[5]; ids[0] = ControlID1 ; ids[1] = HelpID1 ; ids[2] = ControlID2 ; ids[3] = -1 ; ids[4] = 0 ;</pre>

Example

```
HtmlHelp(  
    hwndCTRL,  
    "c:\\myHelp.chm::/popups\\cshelp.txt",  
    HH_TP_HELP_CONTEXTMENU,  
    (DWORD) ids) ;
```


- **HH_TP_HELP_CONTEXTMENU** does not display the **What's This?** **Comments** the exact same thing as **HH_TP_HELP_WM_HELP**.
- **HH_TP_HELP_CONTEXTMENU** is very similar to the WinHelp **HELP_CONTEXTMENU** *uCommand* parameter.

See also

[HH_DISPLAY_TEXT_POPUP](#)

[HH_TP_HELP_WM_HELP](#)

 [About commands](#)

HH_TP_HELP_WM_HELP command

Opens a pop-up help topic. Generally used in response to the Windows **WM_HELP** message. For example, this message is sent when a user presses F1.

<i>hwndCaller</i>	<i>pszFile</i>	<i>dwData</i>
Specifies the window handle of the dialog box control for which you want pop-up help to appear. This is typically the control that has focus.	<p>Specifies the compiled help (.chm) file, and the text file that contains the pop-up help topics.</p> <p>By default, the text file is named Cshelp.txt. If Cshelp.txt is located in the root of the compiled help file, then you only need to specify the help file name. If not, you must also specify the relative path.</p>	<p>Specifies an array of DWORDs containing pairs of dialog box control IDs and help topic IDs. The array must be terminated by zero. If a control does not require pop-up help, specify -1 for the help topic ID.</p> <pre>DWORD ids[5]; ids[0] = ControlID1 ; ids[1] = HelpID1 ; ids[2] = ControlID2 ; ids[3] = -1 ; ids[4] = 0 ;</pre>

Example

```
HtmlHelp(  
    hwndCTRL,  
    "c:\\myHelp.chm::/popups\\cshelp.txt",  
    HH_TP_HELP_WM_HELP,  
    (DWORD) ids) ;
```

- **HH_TP_HELP_WM_HELP** is very similar to the WinHelp **HELP** *uCommand* parameter.

Comments

See Also

[HH_DISPLAY_TEXT_POPUP](#)
[HH_TP_HELP_CONTEXTMENU](#)

 [About commands](#)

HH_UNINITIALIZE command

This command is called to properly shut down HTML Help. This function should be the last help command the application calls.

HH_UNINITIALIZE should not be called during DLL process detach, but during the normal application shutdown process.

<i>pszFile</i>	<i>dwData</i>
Must be NULL.	Specifies a cookie. This is the cookie returned by HH_INITIALIZE .

Example

```
HtmlHelp(  
    NULL,  
    NULL,  
    HH_UNINITIALIZE,  
    (DWORD)dwCookie) ; // Pass in cookie.
```


these commands once during the life of your application.

Comments `_UNINITIALIZE` before or at process detach.

See also

[HH_INITIALIZE](#)

 [About commands](#)

About structures

The HTML Help API provides several objects that you work with using data structures.

When working with a structure, it is recommended that you use a Win32 function such as **ZeroMemory**, **memcpy**, or **memset** to clear out the address space of a structure that has been declared. This ensures that you start with a known quantity before making any modifications. For example, the following code fragment uses **memset** to make a copy of an [HH_WINTYPE](#) structure:

```
HH_WINTYPE hhWinType;           //Create new wintype.

HH_WINTYPE *phhWinType;        //Create a pointer to this wintype.

memset(phhWinType,             //Requires "memory.h" header be included in
0,
sizeof(HH_WINTYPE));

HtmlHelp(hwnd,
"c:\\help\\MyHelpFile.chm>main",
HH_GET_WIN_TYPE,
```

- The Win32 functions such as **ZeroMemory**, **memcpy**, or **memset** do not make a deep copy of the structure. Developer-written copy constructors perform proper deep copies for OOP/C++ classes and objects.

Note

Structures quick reference

- [HH_POPUP](#)

<i>Category</i>	<i>Structure</i>
Window types	<ul style="list-style-type: none">• HH_WINTYPE• HH_AKLINK
ALink name/KLink keyword lookups	<ul style="list-style-type: none">• HHN_NOTIFY• HHNTRACK
Notification messages	<ul style="list-style-type: none">• HH_FTS_QUERY
Full-text search	<ul style="list-style-type: none">• HH_LAST_ERROR
Error tracking	

 [About the HTML Help API](#)

HH_AKLINK structure

Use this structure to specify one or more ALink names or KLink keywords

that are used to search for help topics. If *fIndexOnFail* is TRUE, the **Index** tab is selected in the

help window specified in *pszWindow*, and the keyword specified in

pszKeyword is selected in the entry field.

If the lookup yields no matching topics, `HtmlHelp()` checks the values of

the following **HH_AKLINK** members to determine what alternative action

to take.

- *pszURL*. If *fIndexOnFail* is FALSE, the topic file specified in *pszURL* appears in the help window specified in *pszWindow*.
- *pszMsgText* and *pszMsgTitle*. If *fIndexOnFail* is FALSE and *pszURL* is NULL, a message box appears using the text and caption specified in *pszMsgText* and *pszMsgTitle*.

Used by

HH_AKLINK structure syntax

```
typedef struct tagHH_AKLINK
{
    int    cbStruct;

    BOOL   fReserved;

    LPCTSTR pszKeywords;

    LPCTSTR pszUrl;

    LPCTSTR pszMsgText;

    LPCTSTR pszMsgTitle;

    LPCTSTR pszWindow;

    BOOL   fIndexOnFail;
} HH_AKLINK;
```

Member	Description
<i>cbStruct</i>	Specifies the size of the structure. This value must always be filled in before passing the structure to the HTML Help API.
<i>fReserved</i>	This parameter must be set to FALSE.
<i>pszKeywords</i>	Specifies one or more ALink names or KLink keywords to look up. Multiple entries are delimited by a semicolon.
<i>pszUrl</i>	Specifies the topic file to navigate to if the lookup fails. <i>pszURL</i> refers to a valid topic within the specified compiled help (.chm) file and does not support Internet

	protocols that point to an HTML file.
<i>pszMsgText</i>	Specifies the text to display in a message box if the lookup fails and <i>fIndexOnFail</i> is FALSE and <i>pszURL</i> is NULL. The selected topic, if the lookup yields one or more matching topics.
<i>pszMsgTitle</i>	Specifies the caption of the message box in which the <i>pszMsgText</i> parameter appears. <ul style="list-style-type: none"> • The topic specified in <i>pszURL</i>, if the lookup fails and a topic is specified in <i>pszURL</i>.
<i>pszWindow</i>	Specifies the name of the window type in which to display one of the following: <ul style="list-style-type: none"> • The Index tab, if the lookup fails and <i>fIndexOnFail</i> is specified as TRUE.
<i>fIndexOnFail</i>	Specifies whether to display the keyword in the Index tab of the HTML Help Viewer if the lookup fails. The value of <i>pszWindow</i> specifies the Help Viewer.

- If the lookup yields one or more matching topics, the topic titles appear in the **Comments Found** dialog box.

 [About structures](#)

Use this structure for full-text search.

HH_PTS_QUERY structure

Used by

HH_FTS_QUERY structure syntax

```
typedef struct tagHH_FTS_QUERY
```

```
{
```

```
    int    cbStruct;
```

```
    BOOL   fUnicodeStrings;
```

```
    LPCTSTR pszSearchQuery;
```

```
    LONG   iProximity;
```

```
    BOOL   fStemmedSearch;
```

```
    BOOL   fTitleOnly;
```

```
    BOOL   fExecute;
```

```
    LPCTSTR pszWindow;
```

```
} HH_FTS_QUERY;
```

Member	Description
<i>cbStruct</i>	Specifies the size of the structure.
<i>fUnicodeStrings</i>	TRUE if all strings are Unicode.
<i>pszSearchQuery</i>	String containing the search query.
<i>iProximity</i>	Word proximity.
<i>fStemmedSearch</i>	TRUE for StemmedSearch only.
<i>fTitleOnly</i>	TRUE for Title search only.
<i>fExecute</i>	TRUE to initiate the search.
<i>pszWindow</i>	Window to display in.



[About structures](#)

HH_LAST_ERROR structure

This structure returns the last HtmlHelp() error code and description.

Used by

HH_LAST_ERROR structure syntax

```
typedef struct tagHH_LAST_ERROR
{
    int    cbStruct ;

    HRESULT hr ;

    BSTR   description ;
} HH_LAST_ERROR ;
```

Member	Description
<i>cbStruct</i>	Specifies the size of the structure. This value must always be filled in before passing the structure to HtmlHelp().
<i>hr</i>	Specifies the last error code.
<i>description</i>	Specifies a Unicode string containing a description of the error.

 [About structures](#)

Use this structure to specify or modify the attributes of a pop-up window.

HT_POPUP structure

Used by

HH_POPUP structure syntax

```
typedef struct tagHH_POPUP
{
    int    cbStruct;

    HINSTANCE hinst;

    UINT    idString;

    LPCTSTR pszText;

    POINT   pt;

    COLORREF clrForeground;

    COLORREF clrBackground;

    RECT    rcMargins;

    LPCTSTR pszFont;
} HH_POPUP;
```

Member	Description
<i>cbStruct</i>	Specifies the size of the structure. This value must always be filled in before passing the structure to <code>HtmlHelp()</code> .
<i>hinst</i>	Instance handle of the program or DLL to retrieve the string resource from. Ignored if <i>idString</i> is zero, or if <i>idString</i> specifies a file name.
<i>idString</i>	Specifies zero, a resource ID, or a topic number in a text

	file.
<i>pszText</i>	Specifies the text to display if <i>idString</i> is zero.
<i>pt</i>	Specifies (in pixels) where the top center of the pop-up window should be located.
<i>clrForeground</i>	Specifies the RGB value to use for the foreground color of the pop-up window. To use the system color for the window text, specify -1.
<i>clrBackground</i>	Specifies the RGB value to use for the background color of the pop-up window. To use the system color for the window background, specify -1.
<i>rcMargins</i>	Specifies (in pixels) the margins to use on the left, top, right, and bottom sides of the pop-up window. The default for all rectangle members is -1.
<i>pszFont</i>	<p>Specifies the font attributes to use for the text in the pop-up window.</p> <p>Use the following format to specify font family, point size, character set, and font format:</p> <p>facename[, point size[, charset[BOLD ITALIC UNDERLINE]]]</p> <p>To omit an attribute, enter a comma. For example, to specify bold, 10-pt, MS Sans Serif font, <i>pszFont</i> would be:</p> <p>MS Sans Serif, 10, , BOLD</p>

 [About structures](#)

Use this structure to specify or modify the attributes of a [window type](#).

Window types can be defined by an author in a project (.hhp) file, or they can be defined programmatically using the HTML Help API.

When a **HH_WINTYPE** structure is passed to HtmlHelp() using the **HH_SET_WIN_TYPE** command, the HTML Help API makes a private copy of the contents of the structure. The help developer is therefore responsible for freeing memory used by the **HH_WINTYPE** structure or character arrays within it. The help developer can free memory after calling **HH_SET_WIN_TYPE**.

Used by

HH_WINTYPE structure syntax

```
typedef struct tagHH_WINTYPE
{
    int        cbStruct;

    BOOL       fUnicodeStrings;

    LPCTSTR    pszType;

    DWORD      fsValidMembers;

    DWORD      fsWinProperties;

    LPCTSTR    pszCaption;

    DWORD      dwStyles;

    DWORD      dwExStyles;

    RECT       rcWindowPos;

    int        nShowState;

    HWND       hwndHelp;

    HWND       hwndCaller;

    HWND       hwndToolBar;

    HWND       hwndNavigation;

    HWND       hwndHTML;
}
```

```

int      iNavWidth;

RECT     rcHTML;

LPCTSTR  pszToc;

LPCTSTR  pszIndex;

LPCTSTR  pszFile;

LPCTSTR  pszHome;

DWORD    fsToolBarFlags;

BOOL     fNotExpanded;

int      curNavType;

int      idNotify;

LPCTSTR  pszJump1;

LPCTSTR  pszJump2;

LPCTSTR  pszUrlJump1;

LPCTSTR  pszUrlJump2;

} HH_WINTYPE;

```

Member	Description
<i>cbStruct</i>	Specifies the size of the structure. This value must always be filled in before passing the structure to HtmlHelp().
<i>fUnicodeStrings</i>	Specifies whether the strings used in this structure are

	UNICODE.
<i>pszType</i>	A null-terminated string that specifies the name of the window type.
<i>fsValidMembers</i>	Specifies which members in the structure are valid.
<i>fsWinProperties</i>	Specifies the properties of the window, such as whether it is the standard HTML Help Viewer or whether it includes a Search tab.
<i>pszCaption</i>	A null-terminated string that specifies the caption to display in the title bar of the window.
<i>dwStyles</i>	Specifies the styles used to create the window. These styles can be ignored, combined with extended styles, or used exclusively depending on the value of the <i>fsValidMembers</i> and <i>fsWinProperties</i> parameters.
<i>dwExStyles</i>	Specifies the extended styles used to create the window. These styles can be ignored, combined with default styles, or used exclusively depending on the value of the <i>fsValidMembers</i> and <i>fsWinProperties</i> parameters.
<i>rcWindowPos</i>	Specifies the coordinates of the window in pixels. The values are read in the following order: rcWindowPos = {left, top, right, bottom};
<i>nShowState</i>	Specifies the initial display state of the window. Valid values are the same as those for the Win32 API ShowWindow function.
<i>hwndHelp</i>	Specifies the handle of the window if the window has been created.
<i>hwndCaller</i>	Specifies the window that will receive HTML Help notification messages. Notification messages are sent via Windows WM_NOTIFY messages.
<i>idNotify</i>	Specifies a non-zero ID for enabling HTML Help notification messages. This ID is passed as the <i>wParam</i> value of Windows WM_NOTIFY messages.

Members specific to the [Help Viewer](#)

Member	Description
<i>hwndToolBar</i>	Specifies the handle of the toolbar.
<i>hwndNavigation</i>	Specifies the handle of the Navigation pane.
<i>hwndHTML</i>	Specifies the handle of the Topic pane, which hosts Shdocvw.dll.
<i>iNavWidth</i>	Specifies the width of the Navigation pane when the Help Viewer is expanded.
<i>rcHTML</i>	Specifies the coordinates of the Topic pane.
<i>pszToc</i>	Specifies the contents (.hhc) file to display in the Navigation pane.
<i>pszIndex</i>	Specifies the index (.hhk) file to display in the Navigation pane.
<i>pszFile</i>	Specifies the default HTML file to display in the Topic pane.
<i>pszHome</i>	Specifies the file or URL to display in the Topic pane when the Home button is clicked.
<i>fsToolBarFlags</i>	Specifies which buttons to include on the toolbar.
<i>fNotExpanded</i>	Specifies that the Help Viewer open with the Navigation pane closed.
<i>curNavType</i>	Specifies the default tab to display on the Navigation pane.
	Specifies the text to display

<i>pszJump1</i>	underneath the Jump1 button.
<i>pszJump2</i>	Specifies the text to display underneath the Jump2 button.
<i>pszUrlJump1</i>	Specifies the URL to jump to when the Jump1 button is clicked.
<i>pszUrlJump2</i>	Specifies the URL to jump to when the Jump2 button is clicked.

See Also

 [About structures](#)

Use this structure to return the file name of the topic that has been navigated to, or to return the window type name of the help window that has been created.

Used by

HHN_NOTIFY structure syntax

```
typedef struct tagHHN_NOTIFY
```

```
{
```

```
    NMHDR hdr;
```

```
    PCSTR pszUrl;
```

```
} HHN_NOTIFY;
```

Member	Description
<i>hdr</i>	Standard WM_NOTIFY header.
<i>pszUrl</i>	A multi-byte, zero-terminated string that specifies the topic navigated to, or the name of the help window being created.

 [About structures](#)

This structure returns the file name of the current topic and a constant that specifies the user action that is about to occur, such as hiding the Navigation pane by clicking the **Hide** button on the toolbar.

Used by

HHNTRACK structure syntax

```
typedef struct tagHHNTRACK
{
    NMHDR    hdr;

    PCSTR    pszCurUrl;

    int      idAction;

    HH_WINTYPE* phhWinType;
} HHNTRACK;
```

Member	Description
<i>hdr</i>	Standard WM_NOTIFY header.
<i>pszCurUrl</i>	A multi-byte, zero-terminated string that specifies the current topic (before the action is taken).
<i>idAction</i>	Specifies the action the user is about to take. This is an HHACT_ constant.
<i>phhWinType</i>	A pointer to the current HH_WINTYPE structure.

 [About structures](#)

- The value of *hwndCaller* must be non zero.

To enable notification message tracking

- The value of *idNotify* must be non zero.

HTMAid sends notification messages from **HHWIN_PARAM_PROPERTIES** have the following **HH_WINTYPE** property settings:

- A value must be specified for **HHWIN_PROP_TRACKING** in the *fsWinProperties* member.

To specify the window that will receive notification messages

Notification messages are sent to the window specified in the *hwndCaller* parameter of the `HtmlHelp()` call.

HTML Help notification messages are sent via Windows **WM_NOTIFY** messages.

Note

[HHN_WINDOW_CREATE](#)

Notification messages quick reference

 [About the HTML Help API](#)

HHN_NAVCOMPLETE message

Sent when the user successfully navigates to a topic in a compiled help (.chm) file.

Notification code	IParam
HHN_NAVCOMPLETE	A pointer to an HHN_NOTIFY structure.

messages.

Comments

 [About notification messages](#)

HHN_TRACK message

Sent when a user clicks a button on the toolbar or a tab in the Navigation pane of the HTML Help Viewer. The message is sent before the action is taken.

Notification code	lParam
HHN_TRACK	A pointer to an HHNTRACK structure.

messages.

Comments click results in navigation to a different topic, an [HHN_NAVCOMPLETE](#) message is also sent.

 [About notification messages](#)

HHN_WINDOW_CREATE message

Sent right before a help window is created.

Notification code	iParam
HHN_WINDOW_CREATE	A pointer to an HHN_NOTIFY structure.

messages.

Comments

 [About notification messages](#)

The *fsValidMembers* member is used when modifying an existing window type, and determines which of the other members should be updated. This parameter can be a combination of one or more of the following values:

Value	Description
HHWIN_PARAM_CUR_TAB	The <i>curNavType</i> member is valid.
HHWIN_PARAM_EXPANSION	The <i>fNotExpanded</i> member is valid.
HHWIN_PARAM_EXSTYLES	The <i>dwExStyles</i> member is valid.
HHWIN_PARAM_NAV_WIDTH	The <i>iNavWidth</i> member is valid.
HHWIN_PARAM_PROPERTIES	The <i>fsWinProperties</i> member is valid.
HHWIN_PARAM_STYLES	The <i>dwStyles</i> member is valid.
HHWIN_PARAM_RECT	The <i>rcWindowPos</i> member is valid.
HHWIN_PARAM_SHOWSTATE	The <i>nShowState</i> member is valid.
HHWIN_PARAM_TB_FLAGS	The <i>fsToolBarFlags</i> member is valid.

[← Back to HH_WINTYPE](#)

The *fsWinProperties* member is used to define various properties for a help window, and is also used to control how the window is created.

This member can be a combination of one or more of the following values:

Value	Description
HHWIN_PROP_AUTO_SYNC	Locates and selects the contents or index entry (depending on whether the Contents or Index tab is selected in the Navigation pane) for the help topic that appears in the Help Viewer. If the topic cannot be located in the Navigation pane, no action is taken.
HHWIN_PROP_CHANGE_TITLE	Changes the title bar of the Help Viewer to match the title of the topic that appears in the Topic pane.
HHWIN_PROP_NAV_ONLY_WIN	Specifies that the window only contain a Navigation pane and toolbar. No topics can appear in this Help Viewer.
HHWIN_PROP_NODEF_EXSTYLES	Specifies that the window use no default extended styles when it is created. If used in combination with the HHWIN_PROP_ONTOP flag, the HHWIN_PROP_ONTOP flag is ignored.
	Specifies that the window not use default styles when it is created. If this flag is not specified, the window is created with the

HHWIN_PROP_NODEF_STYLES	<p>following default styles:</p> <p>WS_THICKFRAME WS_OVERLAPPED WS_VISIBLE</p>
HHWIN_PROP_NOTB_TEXT	Specifies that the buttons on the toolbar of the Help Viewer not contain text below the icon.
HHWIN_PROP_NOTITLEBAR	Creates a window without a title bar.
HHWIN_PROP_NO_TOOLBAR	Creates a window without a toolbar.
HHWIN_PROP_ONTOP	Creates a window that stays on top of all windows on the desktop rather than staying on top of only the calling window.
HHWIN_PROP_TAB_ADVSEARCH	<p>Creates a window whose Navigation pane includes a full-text Search tab.</p> <p>If HHWIN_PROP_TAB_SEARCH is not specified as a value for <i>fsWinProperties</i>, no Search tab will appear, whether or not the compiled help (.chm) file includes full-text search information.</p>
HHWIN_PROP_TAB_FAVORITES	Creates a window whose Navigation pane includes a Favorites tab.
HHWIN_PROP_TAB_SEARCH	Creates a window whose Navigation pane includes a Search tab.
	Enables HTML Help notification

HHWIN_PROP_TRACKING	messages to be sent to the window specified in <i>hwndCaller</i> .
HHWIN_PROP_TRI_PANE	Creates the HTML Help Viewer (with a Topic pane, Navigation pane, and toolbar). Once a window has been created, this flag cannot be used to modify the existing window.
HHWIN_PROP_USER_POS	If the user resizes or repositions the help window, this size/position will be used when the compiled help file is opened again. Without this setting, the window always appears in the same size/position.

← [Back to HH_WINTYPE](#)

The `curNavType` member is used to specify which tab in the Navigation pane of the HTML Help Viewer should have focus.

This member can be one of the following values:

Value	Description
<code>HHWIN_NAVTYPE_FAVORITES</code>	Displays the Favorites tab.
<code>HHWIN_NAVTYPE_INDEX</code>	Displays the Index tab.
<code>HHWIN_NAVTYPE_SEARCH</code>	Displays the Search tab.
<code>HHWIN_NAVTYPE_TOC</code>	Displays the Contents tab.

 [Back to HH_WINTYPE](#)

idAction parameter

The *idAction* member can be one of the following values:

Value	Description
HHACT_BACK	The user clicked the Back button.
HHACT_CONTRACT	The user clicked the Hide button.
HHACT_EXPAND	The user clicked the Show button.
HHACT_FORWARD	The user clicked the Forward button.
HHACT_HOME	The user clicked the Home button.
HHACT_JUMP1	The user clicked the Jump1 button.
HHACT_JUMP2	The user clicked the Jump2 button.
HHACT_OPTIONS	The user clicked the Options button.
HHACT_PRINT	The user clicked the Print button.
HHACT_REFRESH	The user clicked the Refresh button.
HHACT_STOP	The user clicked the Stop button.
HHACT_SYNC	The user clicked the Locate button.
HHACT_TAB_CONTENTS	The user clicked the Contents tab.
HHACT_TAB_FAVORITES	The user clicked the Favorites tab.
HHACT_TAB_INDEX	The user clicked the Index tab.
HHACT_TAB_SEARCH	The user clicked the Search tab.
HHACT_ZOOM	The user clicked the Zoom button.

 [Back to HHNTRACK](#)