

MS Help 2 Primer

 web.archive.org/web/20220928222645/http://helpware.net/FAR/help/msh2_primer.htm

This document is designed as a quick introduction to MS Help 2, the new help engine from Microsoft used by VS .NET, MSDN, Borland C# Builder etc.

- MS Help 2 is a completely new help system. New file extensions. New API DLLs.
- MS Help 2 requires a minimum of MS IE5.0 because of its XML requirement.
- Like MS HTML Help 1.x before it, MS Help 2 uses HTML topics, TOC, Index, Search, Favorites, as well as KLinks & ALinks.
- MS Help 2 can have multiple indexes. Think of an index as a list of Name-Value pairs. These lists are static in that they are set at compile time.
- MS Help 2 Data Services allow you to interface with the help file data via COM interfaces (not published by MS).
- Some MS HTML Help 1.x features may not have been implemented in MS Help 2. Note however you can use a mixture of HTML Help 1.x and MS Help 2 in your projects. Remember that some features such as HH 1.x Shortcuts will not work outside of a CHM.
- MS provide a conversion tool HxConv.exe which does a good job of converting .CHM or .HHP over to a MS Help 2 project.
- The compiler no longer strips info from source files. And round trip compiles mean that when you decompile a HxS file you get all the original source files back, including the .HxC project file.
- You must specify the exact list of files to be compiled into target .HxS help file, unlike HH 1.x which could crawl hyper-links to find files. The H2 file list is kept in a .HxF file and may contain wild cards such as "*.htm" and "images*.gif".
- No more confusion over Binary or Non-binary TOCs and Indexes. They are always stored in the help file as binary.
- MS Help 2 collections can be very big. Bigger than MS HTML Help 1.x merged document sets.
- MS Help 2 dynamic filters work on TOC, Index and Search items, and even right down to the paragraph level in a topic file (with some help of JScript). Filters are based on topic attributes embedded in each topic.






Each MS Help 2 collection must be registered before it can be viewed. Registration simply means giving each collection a unique name (text label). This name is called a "Namespace". An MS Help collection path now starts with a Namespace, not a disk directory path as with MS HH 1.x. This fixes some problems, but creates some new problems. You can no longer move or install a help system without registering. An unregistered help system is useless since you cannot view it.






There are two types of collections. Collections that contain just a single .HxS help file, and collections that contain one or more .HxS help files plus a number of uncompiled collection-level files. In the first case, the single .HxS help file is registered under a namespace name and a single help title is added, itself. In the second case, a .HxC collection-level file is registered under a namespace then all .HxS help files making up the collection are registered as Titles under that namespace.

But isn't the .HxC file, the MS Help project file used to build .HxS help files? Yes, it can be both. As a project file it contains information about how to build a .HxS help file. The equivalent in HH 1.x is the .HHP file. The .HxC project file is needed at build time only. The HxC collection-level file defines how all the help files merge at run-time and must be shipped to the customer with the other HxS compiled help files.

Because a collection is now accessed via its Namespace, URLs are now location independent. Thus we can install software on different machines, to different folders, and the same topic URL will work on all machines -- as long as we register each collection using the same Namespace and TitleID(s).

Warning - DTD Files: Note that only HxC project files should contain an XML reference to a DTD file. The DTD file validates the XML syntax at compile time. DTD files are part of the Workshop installation (not the runtime), so it is very important NOT to include a DTD reference in the collection-level files. If this is done your collection would run OK on your development machine, but produce errors on your customers machines.

	HxC	Collection Definition File	Identifies all other key files and compiler options. Defines navigation data monikers.
	HxF	Include File	Lists all files to be included in the compiled HxS help file. Supports *.*; *.HTM, *.HTML type syntax.
	HxT	TOC Definition File	Table of contents definition file.
	HxK	Index Defn File (Internal Name=K)	KLinks. Keyword Index appear in the viewers Index navigation control. Typically named "K".
	HxK	Index Defn File (Internal Name=A)	ALinks. Associative keywords. Typically named "A".

	HxK	Index Defn File (Internal Name=NamedURLIndex)	The Named URL Index defines URLs for the HomePage, DefaultPage etc. Typically named "NamedURLIndex".
	HxK	Index Defn File (Name=F)	The Context window Index contains a list of help mappings as Keyword-URL pairs. Typically named "F".
	HxE	Sample Definition File	Optional file for defining samples.
	HxV	Virtual Topic Definition File(s)	Virtual topic feature provides a way to treat any URL (web, pdf, image etc) as a first class help topic.
	HxA	Attribute Definition File	Defines topic attributes and filter definitions. MS suggest a you omit this file. Doing so will cause all attributes defined in topic file headers to be enabled.
	Other	Topic files, images, CSS etc	Your help content files.

An absolute path to a topic looks like this:

ms-help://namespace/TitleId/internalpath/topic.htm

It looks more like a web URL. After the MS Help protocol "ms-help://" the namespace identifies the collection we want to access. Next the "TitleId" identifies the help file in the collection. The rest of the path is the internal path to the topic in the help file. The rules are similar to those in HH 1.x which used something like *ms-its:path\help.chm::/internalpath/topic.htm*. To link to a file *topic2.htm* in the same internal folder requires a normal relative link ` click me`. Also, absolute paths such as *ms-help://. .*, are understood by Internet Explorer, just as HH 1.x absolute paths were (eg. *ms-its:c:\path\file.chm::/path/topic.htm*).

There are 2 stages to registering a MS Help collection:

- 1) Register the Collection (.HxS or .HxC) file under a unique Namespace (some text label).
- 2) Register one or more help (.HxS) files to the collection. Each help file must be registered under a unique TitleID (some text label). HxS based collections can only contain a single title (itself).

In the case of a single help file collection, the Namespace and TitleID are often the same.

EG. Help file *testhelp.HxS* may have a URL *ms-help://testhelp/testhelp/file.htm*

Here's an interesting example (below). The single slash, takes us to the top of the Help Title (Compiled HxS help file), while the *../fileid* then takes us up and over into an adjacent Help Title called "MyTitle". In this way our internal relative links can be made independent of the Namespace name.

EG. *ms-help:../MyTitle/internalpath/topic.htm*

MS Help 2 also offers a "Plug-in" feature. Here your namespace (a HxC based collection) can be plugged into a Parent namespace (such as MS.VSCC or BORLAND.BDS). If done correctly the plug-in integrates fully with the parent collection -- TOC, Index, Attributes, Filters and Full-text search information.

The full URL for reaching a plug-in is slightly different. Both Parent and Child namespace names must be used in the path.

EG. *ms-help://nsParent/nsChild/Childfileid/Childinternalpath/topic.htm*

EG. *ms-help://MS.VSCC/hw.h2reg/h2reg/html/index.htm*

Note: HxS based collections can not be plugged into and cannot plug into other collections.

The following files can be compiled to a .HxS binary help file by compiling the .HxC file:

Collection-level XML files do not contain DTD statements and must be shipped uncompressed with the .HxS help files. HxF files are not shipped.

- **HxC** - The main XML collection-level file containing info about the entire collection. This is registered to a Namespace name.
- **HxT** - XML collection-level file. This Table Of Contents file contains a list of TitleIds. These Ids are the same ones used to register each help file under the Namespace. This is the way each Help file's TOC is merged at runtime.
- **HxK** - XML collection-level Index files are normally empty of index entries. This sounds useless, however their presence exposes the Index entries at the collection level (and to any collection plugged into).
- **HxA** - Defines the topic attributes you want made visible at the collection level (and any collection plugged into). Note that if you plug into say MS.VSCC then there is no point in redefining attributes already defined in the MS.VSCC collection.
- **HxR/HxQ** - These binary files contain combined index and search terms used to speed up data access. See VSHIK documentation for more info.
- **HxS/HxI** - One or more binary Help file titles are registered under the Namespace. Optionally (a compiler setting) you may ship a separate .HxI Index files (normally stored within the HxS help file).

As stated above it is possible to ship a single .HxS file collection. However this type of collection cannot plug into Microsoft or Borland collections. Borland may allow you to directly register your .HxS help title directly into a Borland collection. However I do not recommend this. Microsoft do not allow this at all.

MS Help 2.x	HTML Help 1.x	WinHelp	Description
.HxC (XML)	.hhp/.col (INI)	.hpj	Project file / Collection file.
.HxS/.HxI	.chm/.chi	.hlp	Compiled help file. HxI is normally inside the HxS help file but can be compiled as a separate file for CDROM disks.
.HxT (XML)	.hhc (Sitemap)	.cnt	Table Of Contents file.
.HxK (XML)	.hhk (Sitemap)	---	Index file. H2 may have multiple indexes.
HTML	HTML	RTF	WinHelp was RTF based.
DExplore.exe	hh.exe	winhelp.exe	Default Help viewer.
hxcomp.exe	hhc.exe	hcrtf.exe	Command line compiler.

Help 2.x collections	Merged CHMs or MSDN collections	Yes	Modular Help
Topic attribute filters	Info Types (did not work)	No	Content filtering.
Data services	No	No	H2 has programmatic access to data via COM (documentation not released by MS).
<mshelp:keyword>	<object> or <meta>	--	Help 2 now uses XML elements instead of object and Meta tags.
<mshelp:link>	<object>	--	Help 2 now uses XML links instead of objects.