Need Help? HTML Help Builder to the Rescue

Doug Hennig

Creating documentation for an application is something most developers dread. However, using West Wind Technologies' HTML Help Builder makes working on a help file almost fun!

In the December 1999 issue of FoxTalk (the predecessor of FoxRockX), I wrote an article about a new tool from West Wind Technologies called HTML Help Builder. Now, more than fifteen years later, it's time to take another look at HTML Help Builder. It's an indispensable part of my toolkit and if you have to create any type of documentation for your applications, whether in CHM, HTML, PDF, or Microsoft Word format, it should be in your toolkit too.

As you may expect, HTML Help Builder has changed a lot over the years. The newest version, 5.0 (currently in beta but should be released by the time you read this), is the biggest upgrade ever, adding a lot of new features and making it easier than ever to create beautiful yet functional documentation. Let's take a look at HTML Help Builder, with special focus on what's new.

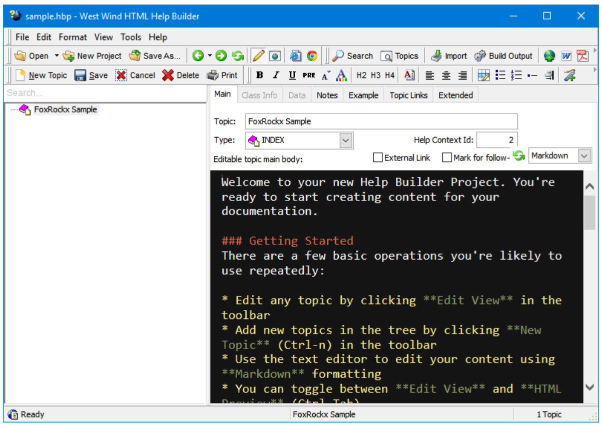
Installing HTML Help Builder

Installing HTML Help Builder is easy: download the shareware version from West Wind's web site (https://helpbuilder.west-wind.com) and install it wherever you wish. The shareware version has all of the features of the registered version but adds a shareware notice to every page of the generated documentation. If you purchase HTML Help Builder (it's only $299 US), you'll receive a registration code to enter into a dialog so there's no need to install anything else.

Using HTML Help Builder

The first step is to create a project for your documentation by clicking the New Project button in the toolbar. The Create a New Project wizard appears, in which you enter the project title, your company name, the directory for the project files (it's best to specify a new directory, since HTML Help Builder creates a lot of files; I usually specify a subdirectory of my application's folder named "HTMLHelp"), and the filename for the main project file (this file, which is really a VFP table, has an HBP extension). After HTML Help Builder creates the project files, it opens the project and you're ready to go.

Figure 1 shows the HTML Help Builder user interface. It consists of a menu bar, several tool bars, a TreeView control showing topics and their hierarchy (initially, there'll only be one topic, which is named the same as the project title), and a display area that either shows a preview of the selected topic ("Topic Html view") or a pageframe with several pages of editable information about the topic ("Topic edit view"). You can switch between HTML and edit view with the appropriate button in the toolbar.


sample.hbp - West Wind HTML Help Builder
File Edit Format View Tools Help
Open New Project Save As... Search Topics Import Build Output
New Topic Save Cancel Delete Print B I U PRE H2 H3 H4
Search...
FoxRockx Sample
Main Class Info Data Notes Example Topic Links Extended
Ready
FoxRockx Sample
1 Topic

Topic: FoxRockx Sample
Type: INDEX Help Context Id: 2
Editable topic main body: □ External Link □ Mark for follow- Markdown

```
Welcome to your new Help Builder Project. You're
ready to start creating content for your
documentation.

### Getting Started
There are a few basic operations you're likely to
use repeatedly:

* Edit any topic by clicking **Edit View** in the
toolbar
* Add new topics in the tree by clicking **New
Topic** (Ctrl-n) in the toolbar
* Use the text editor to edit your content using
**Markdown** formatting
* You can toggle between **Edit View** and **HTML
Preview** (Ctrl-Tab)
```

Figure 1. The HTML Help Builder user interface.

Creating topics

To create a new topic, click the New Topic button in the Toolbar to display the Create New Topic wizard. Select the type of topic to create (I'll discuss types later) and the location to insert the topic: at the same level as the current topic (which is shown in a combobox and can be changed if desired), as a child of the current topic, or at the top level. In step 2, enter the topic title and choose Finish to create the new topic. The new topic appears in the TreeView with a bullet in front of its title, which indicates the topic has no text yet.

Once you've entered text for the topic, the bullet disappears (you may need to click the Refresh Tree button in the toolbar to make this happen right away).

To enter information about the selected topic, select edit view. A pageframe with seven pages appears beside the TreeView. The Main page has the following:

- Topic: the title of the topic. If you change the title, you're asked if cross-links should be updated, meaning HTML Help Builder goes through all topics, looking for links to the former name of this topic and changes them to the new name. I'll discuss cross-links in more detail later.
- Type: I'll discuss types later.
- Help context ID: the ID for the topic you'll use in a HELP ID statement in your application so you can provide context-sensitive help. Although you can manually edit the help context ID, there isn't a need to since HTML Help Builder automatically assigns the next available ID to a new topic.
- External link: turn this setting on and enter a URL if you want the body of the topic to display the content of an external URL.
- Mark for follow-up: turn this setting on to indicate that you're still working on this topic. The generated content has a "under construction" header and the TreeView displays a bullet beside the topic title. You can also search for such topics using the Search dialog. I use this feature all the time so I can quickly return to topics I started but haven't finished.
- Content type: this indicates how you enter formatted text in the body. The choices are Markdown (the default), Help Builder (mostly provided for backward compatibility with projects created in earlier versions), and Html.

Below these settings is the body editor. This is the biggest change in HTML Help Builder version 5: rather than using a VFP editbox as previous versions did, it uses the ACE editor control. This powerful editor provides themes, syntax highlighting, spell checking, search and replace, and other features that make working with it a much richer experience. One downside of the control is that it causes some odd focus issues in a VFP application like HTML Help Builder but they're minor and easily worked around.

The default content type is Markdown. Markdown is a relatively new text markup format that translates a set of simple directives into HTML. The HTML Help Builder help file provides basic information on using Markdown and links to more advanced topics. If you've used earlier versions of HTML Help Builder, you'll find Markdown much easier to work with than Help Builder syntax (which is really just HTML with << and >> instead of < and > for tags). Here's an example using Help Builder syntax:

```
<<h1>>Heading<</h1>>

This is <<strong>>bolded<</strong>> text and
this is <<i>>italicized<</i>>.

<<ul>>
<<li>>Bullet point 1
<<li>>Bullet point 2
<</ul>>
```

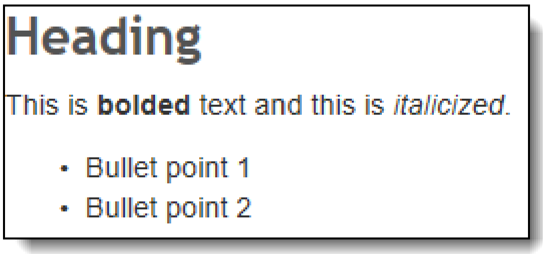
Here's the equivalent in Markdown:

```
# Heading

This is **bolded** text and this is
*italicized*.

* Bullet point 1
* Bullet point 2
```

The rendered text for both is shown in Figure 2.



Heading

This is **bolded** text and this is *italicized*.

- Bullet point 1
- Bullet point 2

Figure 2. The result is the same for Help Builder and Markdown syntax but Markdown is easier to work with.

Of course, you don't have to type directives for simple formatting like bold, italics, and headings; toolbar buttons and shortcut keys provide fast access to many formatting options.

Looking at formatting directives doesn't give you a good idea of what the topic looks like when rendered, so there are a couple of preview options available. One I used to use a lot is switching between edit and HTML view. However, in version 5, the Web Browser preview window has been improved significantly, so now I simply open that window and leave it open. It automatically updates when you move to another topic or save the current one, so there's no longer a need to switch editing modes.

The Topic Links page allows you to do several things. Each topic has a unique ID, which is a SYS(2015) value HTML Help Builder assigns to it and is also used for the name of the HTML file to generate. The Topic Links page shows this ID. It allows you to change the parent topic, which you can also do by dragging the topic in the TreeView and dropping it onto the new parent. You can select related topics from a combobox of topics or, by clicking the magnifying glass button, from a TreeView of topics; topics added to the list appear in a "See Also" item in the generated help. You can enter keywords for the topic which can be used for searching. You can change the order in which the topic appears under its parent topic, although it's easier to do this by choosing Sort Current Topic Tree from the Tools menu or shortcut menu for the TreeView, and moving topics up or down in the dialog that appears. Finally, you can indicate whether this topic appears as a content topic in the HTML Help file.

Images, cross-links, and external links

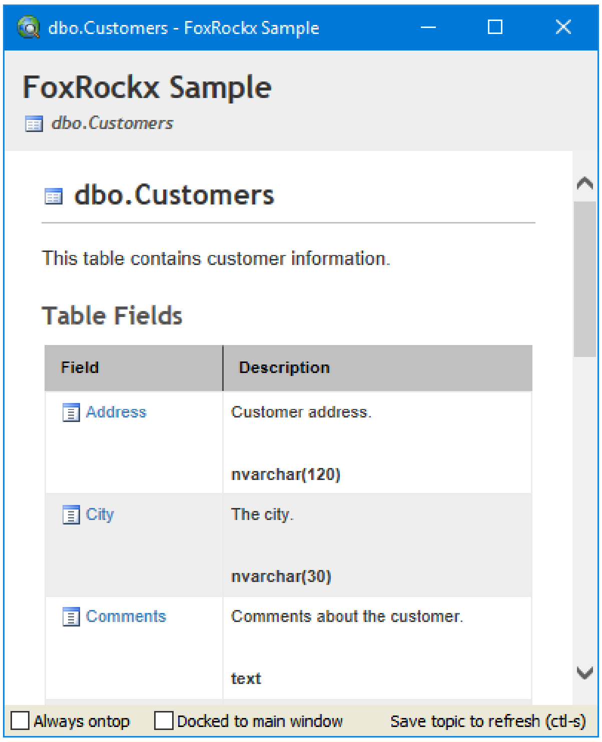
As with web pages, help topics can include images, links to other topics in the help, or even links to web sites. To insert an image into a topic, select edit view, place the cursor where the image should go, click the Insert Image button in the toolbar, and select an image from the dialog that appears. You should place images into the Images subdirectory of the project directory; the HTML Help Compiler seems to have a problem with images in other directories.

Linking one topic to another is easily done in HTML Help Builder. Select the text you want linked and click on the Insert Bookmark button in the toolbar to display the Insert Bookmark dialog. Select the topic to link to from the combobox (if the selected text is similar to a topic title, that title is selected automatically) or click the magnifying glass button for a TreeView of topics. You'll find the selected text replaced with something like <<%= TopicLink([*selected text*],[*ID of topic to link to*]) %>> (Help Builder syntax) or [*selected text*](VFPS://Topic/*TopicID*) (Markdown syntax), which tells HTML Help Builder to insert something like <a href="topicID.htm">selected text</a> into the output.

You can create external links, which are usually used to link to a Web site, but can also link to anything the HTML Help engine (which is essentially Internet Explorer wrapped with functionality specific for CHM files) can work with, like Word documents. As with internal links, select the text you want linked and click on the Create Hyperlink button in the toolbar to display the Insert Hyperlink dialog. Enter the link text (filled in by default) and URL for the link (including "HTTP://" if it's a link to a Web page) and choose OK.

Importing topics

One cool thing about HTML Help Builder is its ability to automatically document classes and databases. The Import button in the toolbar displays the Import Class or Database dialog, where you can choose one or all classes from a VCX or PRG, a COM object or type library for a COM object, or a .NET assembly, or a SQL Server or VFP database. Figure 3 shows the documentation created when I imported the Customers table from the Northwind SQL Server database.



FoxRockx Sample

dbo.Customers

dbo.Customers

This table contains customer information.

Table Fields

Field	Description
Address	Customer address. nvarchar(120)
City	The city. nvarchar(30)
Comments	Comments about the customer. text

□ Always ontop □ Docked to main window Save topic to refresh (ctl-s)

Figure 3. HTML Help Builder can document your databases.

In the case of classes, it creates help topics for each class, property, and method, using the comments for each as the body. The Class Info page has additional information specific for classes, such as the scope, syntax, parameters, and return value for methods, and the scope, syntax, and values for properties. Information entered into these items appears specially formatted in the rendered help. The Extended page allows you to specify the .NET class member signature as well as the values for specific things like Assembly and Namespace.

For a database, each table and field has its own topic, with the caption (if there is one) as the body. The Data page allows you to edit the data type and attributes (such as whether it supports nulls) for the table or field.

The Notes and Examples pages provide editboxes for notes and examples for both classes and databases. Anything entered into these pages appears specially formatted in the rendered help.

Types, templates, and themes

Topic types are an important concept in HTML Help Builder. The pre-defined types are Topic (used for most topics), Index (contains a link to all other topics, laid out like an index), Header (used for topic group headings), WhatsNew (for What's New topics), and several types for class and database documentation (including ClassHeader, ClassMethod, ClassProperty, Database, and DataTable). You can also define your own types by choosing Modify Theme in the View menu to bring up the Help Builder Theme and Template Editor and choosing Add Type.

So, what are types? They're a way to specify the behavior and template for a topic. For example, a Header type appears with a "book" icon in the TreeView while a Topic type appears with a "document" icon. Each type has its own template, which is a text file with a WCS extension in the Templates subdirectory of the project folder. A template contains a mixture of HTML and script code that defines what the HTML page generated from the topic information looks like.

Here's the content of Topic.WCS that generates a topic:

```
<% Layout="~/templates/_Layout.wcs" %>

<header class="content-title">
    <img src="bmp/<%=
TRIM(oHelp.oTopic.Type)%>.gif">
    <%= EncodeHtml(TRIM(oHelp.oTopic.Topic))
%>
</header>

<div class="content-body" id="body">
<%= oHelp.FormatHTML(oHelp.oTopic.Body) %>
</div>

<% IF !EMPTY(oHelp.oTopic.Remarks) %>
<h3 class="outdent" id="remarks">Remarks</h3>
<%= oHelp.FormatHTML(oHelp.oTopic.Remarks) %>
<% ENDIF %>

<% IF !EMPTY(oHelp.oTopic.Example) %>
<h3 class="outdent" id="example">Example</h3>
<%=
oHelp.FormatExample(oHelp.oTopic.Example)%>
<% ENDIF %>

<% if !EMPTY(oHelp.oTopic.SeeAlso) %>
<h3 class="outdent" id="seealso">See also</h3>
<%= lcSeeAlsoTopics %>
<%  endif %>
```

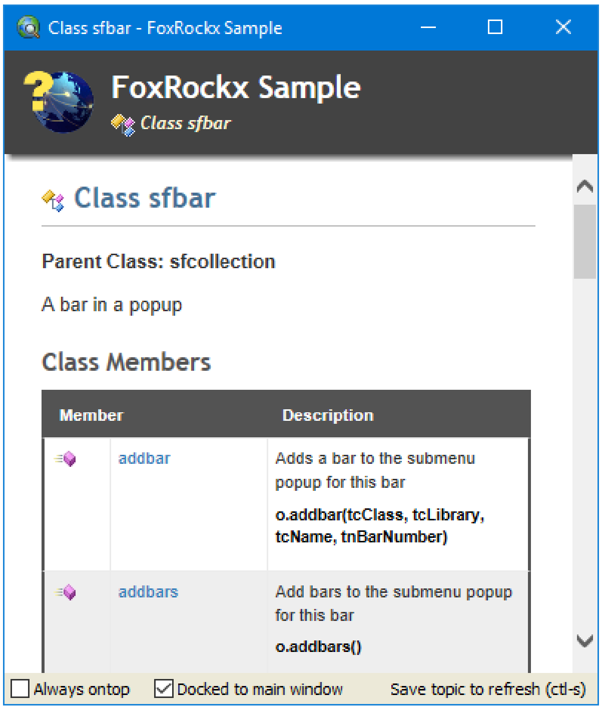
Text between <% and %> delimiters specifies script to be executed rather than text placed into the output. oHelp is a reference to the HTML Help Builder object and oTopic is a reference to the current topic object. So, this template specifies that the majority of the layout comes from _Layout.wcs (another template file), that an image from the BMP folder with the same name as the topic type is displayed followed by the topic title (encoded as HTML in case it contains characters like "&" or ">"), then the rendered body of the topic, followed optionally by remarks, examples, and "see also" text if there is any.

HTML Help Builder comes with different themes. A theme is comprised of HTML templates, CSS, and scripts, and any of those can be changed. These themes affect the appearance of the rendered output. For example, **Figure 4** shows a topic using the Darkhan theme while **Figure 5** shows the Github theme. You can change themes using the Select Theme function in the View menu. Changing the theme causes the contents of the Templates/*ThemeName* folder to be copied to the Templates folder, where the template files for the current theme reside.

If you've used earlier versions of HTML Help Builder, you'll notice the rendered help topics are much more attractive and modern looking in the new version. They also use responsive HTML, thanks to libraries such as Bootstrap, which means the HTML help looks great on any device, from a phone to a tablet to a desktop. To use the new templates for an existing project, choose Options from the Tools menu and click the Update Templates button.

You might think different themes have different templates but in fact they're all the same out of the box. What varies from theme to theme is the style sheet (WWHelp.css) that affects the appearance of the rendered content. Different themes use different color schemes; they could use different fonts, padding, and other things but don't in the provided templates.

HTML Help Builder makes it easy to customize both the templates and style sheet: you can either use the Theme and Template Editor mentioned earlier or functions in the Edit Topic Templates submenu of the View menu.



Class sfbar - FoxRockx Sample

FoxRockx Sample

Class sfbar

Class sfbar

Parent Class: sfcollection

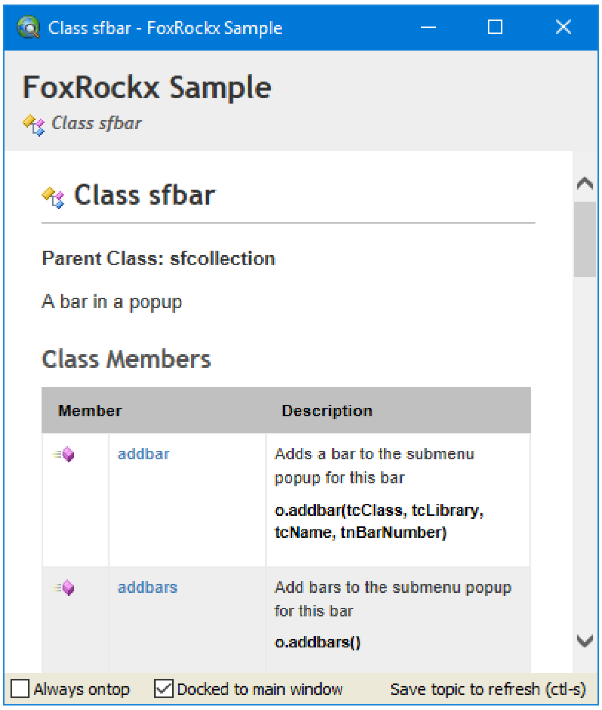
A bar in a popup

Class Members

	Member	Description
◆	addbar	Adds a bar to the submenu popup for this bar **o.addbar(tcClass, tcLibrary, tcName, tnBarNumber)**
◆	addbars	Add bars to the submenu popup for this bar **o.addbars()**

☐ Always ontop ☑ Docked to main window Save topic to refresh (ctl-s)

Figure 4. The Darkhan theme.



Class sfbar - FoxRockx Sample

FoxRockx Sample

Class sfbar

Class sfbar

Parent Class: sfcollection

A bar in a popup

Class Members

	Member	Description
◆	addbar	Adds a bar to the submenu popup for this bar **o.addbar(tcClass, tcLibrary, tcName, tnBarNumber)**
◆	addbars	Add bars to the submenu popup for this bar **o.addbars()**

☐ Always ontop ☑ Docked to main window Save topic to refresh (ctl-s)

Figure 5. The Github theme.

As an example of a custom template, I decided that I wanted to remove the West Wind logos from my rendered topics. I edited the layout template (_Layout.wcs) and removed the following:

```
<img src="images/logo.png" class="banner-logo"
/>
```

and:

```
<div class="pull-right">
  <a href="http://helpbuilder.west-wind.com"
  target="_blank"><img src="images/wwhelp.png"
  /></a>
</div>
```

Creating a help file

When you're ready to create a CHM file, click the Build Output button in the toolbar to bring up the Generate HTML Help wizard. You have a variety of choices for generation, such as where HTML files should be generated and whether only a subset of topics should be created, but I just usually accept all the defaults (which creates a complete CHM file) and click Finish. After a few moments, the Help Compiler Results dialog appears, displaying the status of the build (including any errors you need to correct, such as invalid HTML tags or bad links) and giving you functions to run the help file, run the HTML Help Workshop, view the HTML Help in a browser, or copy the help file to another directory. If everything went well, you should be able to run your help file and see it in action.

If you'd rather generate help as a Word document or PDF, click the appropriate export button in the toolbar (this requires Microsoft Word 2010 or later on your machine). If you haven't already done so, HTML Help Builder reminds you that it works better if you change the theme to MSWORD first.

I like to provide help for my applications as both a CHM file for help within the application and as HTML files on our web site. The reason for the latter is that when someone asks a support question, I can send them a link directly to the applicable help topic (for example, http://stonefieldquery.com/OnlineHelp/EndUser/_1qi0t7qot.htm for the topic describing the Reports Explorer in Stonefield Query). In fact, HTML output is arguably more important than a CHM file; the output from HTML Help Builder is a self-contained web site that can be used for any kind of reference material you can think of: documentation, FAQs, notes, etc. You can either click the FTP Web Upload button in the Help Compiler Results dialog or click the upload button in the toolbar to upload the HTML files for your project to a web site. If you don't need a CHM file, you can skip creating it altogether by choosing Don't Build Help File in the Generate HTML Help wizard.

Summary

West Wind Technologies' HTML Help Builder is one of those "no brainer" purchases: if you or someone on your development team creates CHM, HTML, Word, or PDF-based documentation, it just makes sense to use this tool rather than doing things the hard way. I've been using HTML Help Builder for more than 17 years and couldn't imagine creating an application without it.

Doug Hennig is a partner with Stonefield Software Inc. He is the author of the award-winning Stonefield Database Toolkit (SDT); the award-winning Stonefield Query; the MemberData Editor, Anchor Editor, and CursorAdapter and DataEnvironment builders that come with Microsoft Visual FoxPro; and the My namespace and updated Upsizing Wizard in Sedna.

Doug is co-author of "VFPX: Open Source Treasure for the VFP Developer," "Making Sense of Sedna and SP2," the "What's New in Visual FoxPro" series (the latest being "What's New in Nine"), "Visual FoxPro Best Practices For The Next Ten Years," and "The Hacker's Guide to Visual FoxPro 7.0." He was the technical editor of "The Hacker's Guide to Visual FoxPro 6.0" and "The Fundamentals." All of these books are from Hentzenwerke Publishing (http://www.hentzenwerke.com). He wrote over 100 articles in 10 years for FoxTalk and has written numerous articles in FoxPro Advisor, Advisor Guide to Visual FoxPro, and CoDe. He currently writes for FoxRockX (http://www.foxrockx.com).

Doug spoke at every Microsoft FoxPro Developers Conference (DevCon) starting in 1997 and at user groups and developer conferences all over the world. He is one of the organizers of the annual Southwest Fox conference (http://www.swfox.net). He is one of the administrators for the VFPX VFP community extensions Web site (http://vfpx.codeplex.com). He was a Microsoft Most Valuable Professional (MVP) from 1996 to 2011. Doug was awarded the 2006 FoxPro Community Lifetime Achievement Award (http://tinyurl.com/ygnk73h).