# Developing AppBuilder 3.2 Help

This document discusses how to create AppBuilder 3.2 Help using the confluence pages as source files.

The document includes the following chapters:

* Creating HTML Help Systems
* Generic Plan for Developing AppBuider 3.2 Help

## Creating HTML Help Systems

To create an HTML Help (.chm file) from HTML topics, it is possible to use Microsoft HTML Help Workshop.

This chapter covers the following subjects:

* Downloading Microsoft HTML Help Workshop
* Creating an HTML Help project

##### Downloading Microsoft HTML Help Workshop

You can download a free version of Microsoft HTML Help Workshop 1.32 from Microsoft Download Center using the following link:

<http://www.microsoft.com/en-us/download/details.aspx?id=21138>

**Supported operating systems**: Windows 7, Windows Server 2003, Windows Server 2008, Windows Vista, Windows XP, and some other Windows versions, i.e. it can be used on the TeamCity builder machines that run Windows 7.

The software package includes the following executables:

* **flash.exe** (HTML Help Image Editor): Lets you edit the image files.
* **hhw.exe** (HTML Help Workshop snap-in): Lets you create and edit HTML Help project files.
* **hhc.exe** (HTML Help Compiler): Performs an unattended building of HTML Help (.chm file) from HTML topics.

##### Creating an HTML Help project

An HTML Help project includes HTML Help topic files (.html), the Help project file (.hhp), the Help contents file (.hhc), and the style files (.css). It can optionally include image files (.png, .jpg, etc.) and index file (.hhk).

After you create the project file, you should add to it the locations of your topic, contents, index, and other files. Once your project file is complete, you compile it to create the final help file (.chm).

**To create an HTML Help project**

1. Create appropriate HTML topics.  
   *The HTML topics can be created from scratch or exported from the confluence pages.*
2. Create the Help contents file (.hhc) that contains the topic titles for your table of contents.  
    *This file can be manually created using the HTML Help Workshop snap-in.*
3. Optionally, create the Help index file (.hhk) that contains the index entries (keywords) for your index.  
   *This file can be manually created using the HTML Help Workshop snap-in.*
4. Create the HTML Help project file (.hhp) and then add to it the locations of your topic, contents, index, and other files.  
   *The HTML Help project file is a text file that can be modified to customize the Help windows.*

**To compile an HTML Help project**

* At command prompt, type the following command and press ENTER:  
  hhc.exe *<Help\_Project\_file>*  
  where the <Help\_Project\_file> refers

The following chapter provides a plan for developing AppBuilder 3.3 Help. This plan details the steps of the above procedures.

## Generic Plan for Developing AppBuider 3.2 Help

This chapter provides a draft of the plan for developing AppBuilder 3.2 Help.

The following table summarizes the steps that must be performed to create AppBuilder32 Help from the existing confluence pages available at <https://wiki.bphx.com/display/AB32/Home>. The steps are related with procedures from the Creating an HTML Help project chapter.

Used colors:  
Step name – done  
Step name – in progress  
Step name – to be done

|  |  |  |
| --- | --- | --- |
| **STEP** | **DETAILS** | **REMARKS** |
| Exporting the confluence pages to HTML topic files. | Confluence 3.4 provides the *HTML Export* feature that lets us export selected pages, or the entire contents of the confluence space into a zipped archive of HTML files. This archive also includes the style and image files. | It takes 5-7 minutes to export all confluence pages from AppBuilder32 Help.  *The use of automatic plug-ins is not efficient, because the Help content is rarely updated. In addition, after exporting the confluence pages to HTML topics, as a rule, it is necessary to update the Help contents and style files.* |
| Modifying the Help styles file **site.css** | The site.css file keeps track of design and format information such as the colors, fonts, font sizes, and margins in HTML topics. After exporting the confluence pages to HTML, it is necessary to update this file to create an appropriate design for Appbuilder32 Help. | This file is automatically created after exporting the confluence pages to HTML.  *In this file, we should updates some classes to hide unnecessary elements, change fonts, etc.* |
| Creating the Help project (.hhp) file Appbuilder32.hhp | This file can be created using standard template files available in the HTML Workshop documentation. It is necessary to add the actual locations of topics, image files, etc. to this file. | This file has been already created for current version of AppBuilder32 Help. It will change rarely. |
| Creating the Help contents (.hhc) file. | This file can be created only manually, because it determines the Help system’s look and feel. | The contents file can be created using HTML Help Workshop snap-in.  In HTML Help, I propose to use the current structure of topics in the Appbuilder32 confluence space, but this structure can be changed, if necessary.  *It might take several days to create the contents file for all topics. Actually, it includes only Getting Start Guide and Installation Guide.* |
| Creating the Help index (.hhk) file. | This file can be created only manually. | The index file can be created using HTML Help Workshop snap-in. The list of key words should be discussed with the AppBuilder team. *This file is optional. Actually, it is empty.* |
| Testing the AppBuilder32 Help project on a local PC. | To test the Help project, on a local PC, it is necessary to run the following command: hcc.exe AppBuilder32.hhp where *AppBuilder32.hhp* is the actual name of the current project file. After this, it is necessary to ensure that the AppBuilder32.chm file has been created and can be opened and viewed. | This test is successful. |
| Testing the AppBuilder32 Help project on the builder PC. | This test is similar to the previous test on a local PC. | The tests had been performed on the us-buildw7 and us-tc-ab32-ba1 machines. The both tests were successful. |
| Putting project files under source control | All files that belong to the Appbuilder32 Help project, must be under SVN control. | After discussions with Dmitry Bolshakov, it is recommended to put all necessary files into the following location: svn://subversion/appbuilder/trunk/documentation/HTML  Current status: Awaiting for SVN access. |
| Adding the Help creation step to the AppBuilder building procedure (in TeamCity) | A new step that creates the AppBuilder32.chm file must be added to the AppBuilder32 building procedure. | To be discussed with Dmitry Bolshakov. |

(see next page!)

##### About AppBuilder Context Help pages

The above Table describes generic steps to be performed to build AppBuilder Help from existing confluence pages.

To create the context Help pages that open from AppBuilder API, we should create new HTML topic pages perform the following steps:

1. Create a list of AppBuilder dialogs for which we must create context Help pages.
2. When developing AppBuilder GUI, the developers must make appropriate changes to map each dialog to corresponding Help topic.   
   As a rule, they must activate a special method in the C++ class that implements the dialog, and then pass the dialog ID to Technical Writer. This method opens the appropriate topic in AppBuilder32.chm if the user clicks the **F1** or **Help** button on the dialog.
3. For each dialog, create HTML Help pages (these pages will describe the dialog control elements).  
   Each HTML page must have an appropriate name obtained from the AppBuilder team (see Step 2).
4. The created HTML pages must be added to AppBuilder HTML project and compiled into AppBuilder32.chm using the generic procedure (see Steps in Generic Plan for Developing AppBuider 3.2 Help).

**NOTE**: To create Context Help, I need detailed information on existing dialogs from Dmitry Gorsky. We also must discuss the list of dialogs (it is desirable to provide screenshots or let me access resource files in your API projects with Microsoft Visual Studio). Before this, it is difficult to estimate the time required to perform this work.

##### 