Generating DocBook Documentation for the Quark Project

Neil Corkum
Andrew Eisenberg
Edward Lam

Generating DocBook Documentation for the Quark Project

by Neil Corkum, Andrew Eisenberg, and Edward Lam

Last modified: October 16, 2007

What you need to know to edit files (it's really not all that hard or scary)

Actually, it *is* a little hard and scary, but don't worry, this document is here to help.

1. How to organize the DocBook source files

Create the folder structure as shown in *Figure 1*

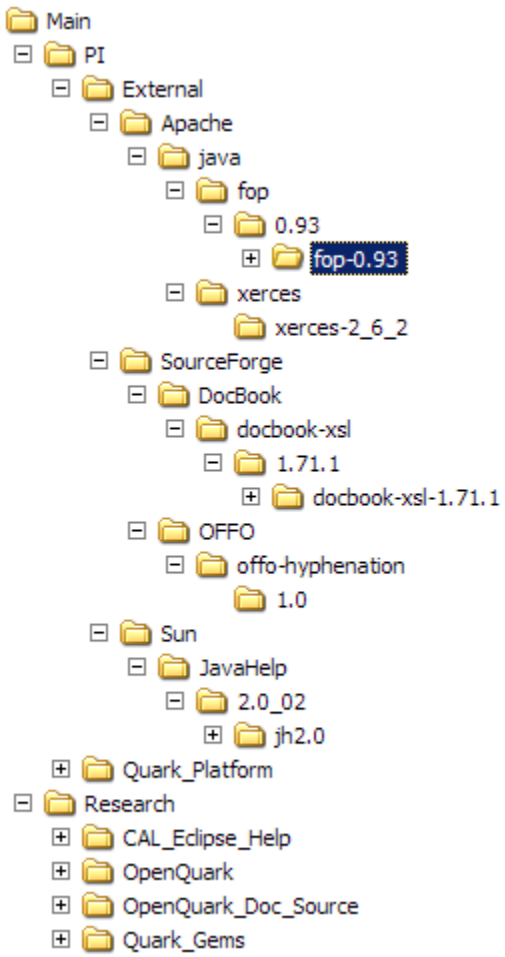

Figure 1. DocBook Source File Structure

Locations of various files are as follows:

Build Scripts
`/OpenQuark_Doc_Source/DocBuild_Scripts/`

Source Files
`/OpenQuark_Doc_Source/xml/`

Generated PDF
`/OpenQuark/docs/`

Generated GemCutter Help
`/Quark_Gems/Help/GemCutterHelpFiles.jar`

Eclipse Help
`/CAL_Eclipse_Help/`

2. How to edit the DocBook files

DocBook is a language based on XML. It gets compiled into either HTML or PDF using XSLT. Lots of fun.

1. See above on how to organize the source files and tools.
2. Download **XMLMind XML Editor** from:

   `http://www.xmlmind.com/xmleditor/download.shtml`
3. Get the **DocBook 5 With XInclude Configuration** add-on:

   `http://www.xmlmind.net/xmleditor/_download/docbook5xi_config-3_6_0.zip`
4. Check out the files you will be editing.
5. Open the files in XMLMind and edit away!

   Just be warned that editing XML in XMLMind is similar to, but not quite like editing HTML in your favorite HTML editor.
6. Add any new images to the appropriate sub-folder of: `/OpenQuark_Doc_Source/xml/images/`

3. How to build the DocBook files

This is the scary part. There are lots of little steps, so read carefully and pay attention. Maybe you want to get another cup of coffee before you read this.

1. Run the build script located at:

   `/OpenQuark_Doc_Source/DocBuild_Scripts/buildall.bat`

   Before running it, you may want to edit it and comment out (**@rem**) the batch files that you do not want called (e.g. you can comment out `GemCutterJavaHelp.bat` if you have not made changes to the GemCutter help). This will prevent unnecessary files from being generated.

2. If the build scripts are throwing Java Exceptions, then it is likely that the document is not a valid XML document.

   Type **Ctrl-Shift-V** to validate. If this doesn't tell you where your problem is, then you are allowed to cry.

3. If you edited Eclipse Help files, there are some more steps.

   a. The help files were generated in a sub-folder below where the script was executed.

   b. Run the script `stripDoctypeDeclarations.bat` located in the `DocBuild_Scripts` folder.

      Note that this script uses cygwin's '**sed**' utility, so ensure that **sed** is on the path.

   c. Move the files from this sub-folder to `/CAL_Eclipse_Help/`

   d. Copy any new images for the eclipse help files from the source directory to the build directory. The build script does not copy them automatically.

4. You may want to review any files you generated in Acrobat and/or a web browser.

4. Handy DocBook reference:

DocBook is a bit like HTML (both are XML/SGML based), but they are very different languages. In general, DocBook translators are less tolerant of errors than browsers are. Here are some good places to read up on DocBook:

DocBook 5 online guide:
`http://www.docbook.org/tdg5/en/html/docbook.html`

DocBook XSL online guide:
`http://www.sagehill.net/docbookxsl/index.html`

DocBook XSL Stylesheet reference online documentation:
`http://docbook.sourceforge.net/release/xsl/current/doc/`

5. That's it!

Enjoy another cup of coffee (or a beer). You deserve it!

The rest of this document describes more about the components in the DocBook toolchain

Components in the DocBook Toolchain

1. Xalan-J

Xalan-J is the XSLT processor used to transform the input DocBook XML documents to intermediate or final form.

The website for Xalan-j is here: *http://xml.apache.org/xalan-j/*

2. Apache FOP

Apache FOP (Formatting Objects Processor) is an open source implementation of an XSL formatting objects (XSL-FO) processor. Its website is located at *http://xmlgraphics.apache.org/fop/*.

To generate PDF documentation, the input DocBook document is first transformed to intermediate XSL-FO input using the DocBook stylesheets. This intermediate output is then consumed by the XSL-FO processor to produce PDFs. Note that since the intermediate form is an XML document representing output document layout, alternative output forms from FOP are theoretically possible, but at the time of this writing only the PDF output implementation is mature.

3. JavaHelp

JavaHelp is an HTML-based help system from Sun which can be used to provide online help for Java applications. Its website is here: *http://java.sun.com/products/javahelp/*.

Generating JavaHelp from the XML source documents involves several steps. First, the XSLT processor must be run on the input DocBook document using the DocBook stylesheets. This will create a large collection of HTML files, along with some other files that are necessary for JavaHelp to function correctly.

One of these files is a generated helpset file with a default name -- this is renamed with an appropriate name.

The next step is to generate the full-text search database for the JavaHelp. This is necessary to make the **Search** functionality in the help work. To generate this, the jhindexer tool that is included in the JavaHelp software package must be used.

Finally, all the files needed by JavaHelp must be packaged into a JAR file. The directory structure in the JAR file should look like this:

Subdirectory	Contents
JavaHelp	`<helpsetname>.hs`, `jhelpidx.xml`, `jhelpmap.jhm`, `jhelptoc.xml`, HTML files for each help page
JavaHelp/images/<filename>	Images displayed in the help referenced from `<filename>.xml`
JavaHelp/JavaHelpSearch	Files generated by jhindexer

