

DocBook Framework (DBF)

The Apache Velocity Developers



V 1.0

Copyright © 2006-2007 The Apache Software Foundation

Table of Contents

| | |
|---|----|
| 1. Preface | 1 |
| 1.1. About this Project | 1 |
| 1.2. License Information | 1 |
| 1.3. Author Information | 1 |
| 2. Introduction | 2 |
| 2.1. Why another framework for rendering docbook? | 2 |
| 2.2. What you need | 3 |
| 2.3. Caveat Emptor! | 3 |
| 3. Using the Framework | 4 |
| 3.1. How to set up your documentation files | 4 |
| 3.2. Customizing your documentation file layout | 5 |
| 3.3. Writing your documentation | 6 |
| 3.4. Notes | 7 |
| Changing the paper size | 7 |
| Referencing images | 7 |
| Adding a new DocBook file to your documentation build | 8 |
| 4. Developer information | 9 |
| 4.1. ant files | 9 |
| 4.2. DocBook reference files | 9 |
| 4.3. XML Resolver | 9 |
| 4.4. Docbook Source files | 9 |
| 4.5. Stylesheets and Driver files | 10 |
| 4.6. StyleSheet customizations | 10 |
| 4.7. PDF StyleSheet information | 11 |
| 4.8. Titlepages | 11 |
| 5. Acknowledgements | 12 |

1. Preface

1.1 About this Project

This project started out as a framework to render documentation for the Apache Velocity project (<http://velocity.apache.org/>) and ended somehow up to be a generic framework to render DocBook documents using Java and driven by Apache ant.

While DocBook format seems to be ubiquitous these days, to our surprise there were not many generic frameworks around that could render all kinds of formats, are platform independent, do not require lots of infrastructure installed and are easily customizable.

Projects either use heavily customized and hacked style sheets or a mix of Java and other applications. Adjusting such a rendering framework to the needs of the Apache Velocity project was not easy, so at some point, we decided to redo this (almost) from scratch.

1.2 License Information

Copyright © 2006-2007 The Apache Software Foundation.

Licensed under the Apache License, Version 2.0 (the "License") you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3 Author Information

This framework and documentation was written by the Apache Velocity Developers. If you have questions, found a bug or have enhancements, please contact us through the Apache Velocity Development Mailing list at [<dev@velocity.apache.org>](mailto:dev@velocity.apache.org)

2. Introduction

2.1 Why another framework for rendering docbook?

The Velocity project used a simple HTML based format called *XDOC* for its documentation for a very long time. However, *XDOC* is not really popular outside the Apache world¹, it renders somehow into HTML but no other formats (unless you consider a set of alpha and beta-level plugins for maven-1 and maven-2) and tool support for this format is not really there.

When an XML based format for documentation is considered, DocBook seems to be a natural choice. So we decided to take a stab at rendering the existing Velocity Docs that are end-user specific (Users Guide, Developers Guide, Reference and the likes) through DocBook.

What we wanted to have, was a framework, that...

- ...renders multiple documents into multiple formats with an uniform look without having to copy a large number of stylesheets, images and other supporting files around.
- ...separates the render framework and the actual documentation to render. It should be sufficient to install the framework only once and then reference it.
- ...uses the standard DocBook XML and XSL zip files available for download. Many of the open source DocBook frameworks use heavily hacked versions and we want to be able to keep up with releases without having to patch the released files every time.
- ...uses current versions of the DocBook reference files, the libraries and supporting tools.
- ...render all formats without connecting to the Internet. Using the Apache XML resolver, it should be possible to use the framework completely standalone. See <http://xml.apache.org/commons/components/resolver/resolver-article.html> for an explanation.
- ...has some documentation so you understand what happens when a format gets rendered and how.
- ...that can be customized easily (if you consider customizing complex XSL style sheets 'easy').
- ...that is platform independent and uses 100% pure Java. No external programs should be needed or called.
- ...that is driven by Apache ant and could be easily embedded into larger builds.

¹And not even in the Apache world...

2.2 What you need

- A Java Runtime. All testing has been done using the Sun JSDK 1.5.0
- Apache Ant version 1.6 or better. The build script uses the macrodef task which was introduced in ant 1.6. Any later version should work, too. Get it from <http://ant.apache.org/>
- The Sun JAI libraries. Please see the README.FIRST file on how to get and install these.

Everything else needed should be included in this package.

2.3 Caveat Emptor!

This framework has been written for the Velocity documentation and we also tried to do a reasonably good job in documentating it.

In any case, the last and final word is in the Subversion repository for the DocBook Framework at <http://svn.apache.org/repos/asf/velocity/docbook/trunk/>

The reference on how to setup and build documentation is the Velocity documentation at <http://svn.apache.org/repos/asf/velocity/docs/> and also the DocBook Framework documentation itself which is located in the docs/ subfolder of the distribution. If in doubt, please check there on how the framework is used.

3. Using the Framework

3.1 How to set up your documentation files

Writing documentation is not just writing text. Often, an author wants to add images, customize the layout of the pages or use specific style information to format documentation in e.g. HTML format. All the required files must be found by the DocBook Framework for creating output files.



- ❶ ant build file
- ❷ custom settings for your build
- ❸ Docbook sources
- ❹ Custom styles for PDF
- ❺ Custom styles for HTML
- ❻ CSS files for HTML
- ❼ Image files for PDF/HTML

Figure 3.1. Recommended layout for a documentation project

It is possible to customize this file layout further to adjust it to existing documentation. If you start a new documentation project, then we recommend that you start with this layout until you are familiar on how the DocBook Framework behaves.

3.2 Customizing your documentation file layout

Unless you absolutely want to change the default settings for building your documentation, you only need to put a single property into the *project.properties* file.

```
dbf.basedir = <path to your DocBook Framework installation>
```

Figure 3.2. Minimum *project.properties* file

The following additional settings can be changed inside the properties file. Except paper type (see below), these settings normally do not need to be changed:

Table 3.1. DocBook Framework properties

| property name | default value | property function |
|-----------------|---------------------|---------------------------------|
| paper.type | Letter | Paper output size for PDF docs |
| src.dir | \${basedir}/src | docbook and related sources dir |
| style.src.dir | \${src.dir}/styles | custom styles directory |
| docbook.src.dir | \${src.dir}/docbook | docbook files directory |
| images.src.dir | \${src.dir}/images | images location |
| css.src.dir | \${src.dir}/css | css files location |
| target.dir | \${basedir}/target | output directory |
| tmp.dir | \${target.dir}/tmp | temporary files location |

If you do not want to use an absolute location for the `dbf.basedir` property (e.g. because you want to check the documentation into a version control system and do not want to update the file all the time depending on who checks this file out where), you can put the DocBook Framework in a subdirectory of your documentation.

If you use Subversion, you can even use the `svn:externals` setting to do this automatically:

Add the following line to the `svn:externals` property of your documentation root

```
docbook http://svn.apache.org/repos/asf/velocity/docbook/trunk
```

and use the following `dbf.basedir` setting¹:

```
dbf.basedir = ${basedir}/docbook
```

¹This also ensures that everytime you check out your documentation, you will get the latest version of the DocBook Framework.

To render your documentation files, you should write a simple ant build file which calls the framework using the `docbook.dir` and `docbook.file` properties. If your docbook file is located in `src/docbook/manual/ToolManual.xml`, your ant build file looks like this:

```
<project name="dbf-docbook" default="all" basedir=".">
  <property file="project.properties"/>

  <target name="all" description="Build documentation">
    <ant antfile="${dbf.basedir}/build-docbook.xml" target="all">
      <property name="docbook.dir" value="manual"/>
      <property name="docbook.file" value="ToolManual"/>
    </ant>
  </target>
</project>
```

Figure 3.3. Sample ant build file for rendering documentation

The resulting documentation file will be located in subdirectories of the `target/manual` directory.

3.3 Writing your documentation

Your DocBook source files normally reside in subdirectories below the `src/docbook` folder. Each document has its own folder that is referenced through the `docbook.dir` property as shown above.

In the example above, running `ant all` (or just `ant`) will build all the documentation formats for the `ToolManual` DocBook file.

Table 3.2. Default formats built by the DocBook Framework

| | |
|------------|--|
| pdf | Adobe PDF format |
| html | Multiple HTML files, one file for each section |
| htmlsingle | One big HTML file |

Both of the HTML format directories will also contain a Zip file suitable for distribution, which contains all HTML files, images and optional CSS files.

3.4 Notes

Changing the paper size

The DocBook Framework renders the pages of the PDF output by default in *US Letter* format (8.5 x 11 inches). This allows printing the resulting PDF in both Letter and A4 format.

If you want to reformat the PDF documentation in A4, you can use the `paper.type` property when invoking `ant` or by setting it permanently in the `project.properties` file.

```
ant -Dpaper.type=A4 will render the documentation in A4.
```

Figure 3.4. Rendering documentation in A4 format

Referencing images

While the docbook files are located in their respective subdirectories below `src/docbook`, your image files should be put into the `src/images` directory.

When writing documentation, images are referenced as `images/<your image file name here>` because this is where they will end up when rendering your documentation.



Warning

If your DocBook writing tool does not allow you to specify image locations, it might not be able to locate the images from `src/images` and just display a broken image symbol. If this concerns you, you can create a symbolic link inside the source directory where your DocBook files reside to the `src/images` directory.

Adding a new DocBook file to your documentation build

Create a new subdirectory inside `src/docbook`. This is where your new DocBook document will reside.

In your documentation ant build file, you must then add a reference to render your new document. To add a DocBook document called *NewGuide.xml* which has been located in the `guide` subdirectory, see the following example:

```
<ant antfile="build-docbook.xml" target="all">
  <property name="docbook.dir" value="guide"/> ❶
  <property name="docbook.file" value="NewGuide"/> ❷
</ant>
```

- ❶ The new DocBook file is located in `src/docbook/guide`.
- ❷ This is the name of the main docbook file *WITHOUT* the ending. The framework will add `.xml` when opening the DocBook file automatically.

Figure 3.5. Adding a new DocBook document

When you add a new document to the framework, you should make sure that it references DocBook DTD files which can be resolved locally. Included are the DTD files for DocBook 4.4, so your document declaration should be

```
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
    "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
```

Figure 3.6. Recommended DTD for DocBook documents.

If you use a different doctype definition, the DocBook Framework will still render your documents, but it will have to connect to the Internet to retrieve the definition files every time you run the build process.

4. Developer information



Important

First take a look at the `MANIFEST` file in the root directory to get an idea what is in this package and what the various files are supposed to do.

4.1 ant files

The `build.xml` file in your documentation directory contains only the driver targets for rendering the documentation. The actual work is done through targets defined in the `build-docbook.xml` ant file in the DocBook Framework.

This file normally should not be changed! If you have to, please let us know, so we can incorporate your changes and/or bug fixes into the main distribution.

`build-docbook.xml` contains three main targets: `pdf`, `html` and `htmlsingle`. Each is responsible for rendering a format. If you want to add another format, please style your new target similar to these.

All default settings are kept in the `docbook.properties` file in the root directory. There should be no need to change these properties, they can be customized in your project directory by using a `project.properties` file.

4.2 DocBook reference files

We use the DocBook XML and XSL distribution archives without any changes to them. The reference files are located in the `src/zip` folder and are expanded into the `target/` directory before the rendering process.

The file names must be reflected in the `docbook.xml.version` and `docbook.xsl.version` properties in the `docbook.properties` file. If you want to use e.g. a newer XSL version, you can put it into `src/zip` and update the `docbook.properties` to reflect this change. Let us know how it works out for you.

4.3 XML Resolver

The framework uses the Apache XML commons resolver to avoid accessing the Internet for Catalog files. The resolver is configured through the `CatalogManager.properties` and `xml-catalog.xml` files in the `src/resolver` directory of the distribution.

If you update e.g. the Docbook XML version, you must also update the catalog file to match the new version. Else the rendering process will have no knowledge of your changes and access the Internet to download the required DTD files.

4.4 Docbook Source files

The sources for each DocBook document to render should be in subdirectories of `src/docbook`. Each document has its own subdirectory and gets rendered separately. Adding a new document is described in the *Adding a new*

DocBook file to your Documentation build note above.

4.5 Stylesheets and Driver files

For each of the formats used by the framework, a stylesheet driver file exists in the DocBook Framework in `src/styles`. These files are `pdf.xsl`, `html.xsl` and `htmlsingle.xsl`.

The driver files are intended to reference the actual style sheet customization and to add some framework specific elements through filtering. This two step process has been chosen because *html* and *htmlsingle* are very similar and it makes no sense to maintain two sets of stylesheet customizations that are virtually identical.

Before usage, these files are copied to `target/tmp` using an ant filter set. This allows you to use the following replacements in the driver files:

Table 4.1. ant filter tokens in the stylesheet customization files

| Filter token | Default Value | Token function |
|---|--------------------------------------|--|
| @file.prefix@ | file:// (Unix) file:/// (Windows) | Prefix for loading a file through the XSL processor. |
| @docbook.xml@ | (computed at runtime) | Location of the DocBook XML files |
| @docbook.xsl@ | (computed at runtime) | Location of the DocBook XSL style sheets |
| @src.dir@ | \${basedir}/src | Location of the source files (DocBook, Images etc.) |
| @tmp.dir@ | \${basedir}/target/tmp | Directory for temporary (scratch) files |
| @<type>.target.dir@ (type is pdf for PDF, html for multi-page HTML and htmlsingle for single-page HTML) | (computed at runtime) | Points to the target directory into which the document is rendered |

Please refer to the provided driver files in `src/styles` in the DocBook Framework on how to use the filter set.

4.6 StyleSheet customizations

You can customize the stylesheets used to render the documentation by adding style files to your project.

The *html* and *htmlsingle* render process uses the same set of customizations, so there are only two possible locations, one for PDF and one for HTML.

The files are located in your project under `src/styles/pdf` and `src/styles/html` respectively. The DocBook Framework only loads a file named `custom.xsl` file from the directory, which in turn can load additional files.

4.7 PDF StyleSheet information

In the footer, the `<releaseinfo>` and `<productname>` elements of the DocBook document are displayed. Each document should have these fields defined.

4.8 Titlepages

Similar to the style sheet customizations, the DocBook Framework can use custom title pages when rendering PDF and HTML output.

The title page format files are also kept in the `src/styles/html` and `src/styles/pdf` sub-directories of your documentation project. If no custom page is given, a default title page is used.

5. Acknowledgements

DocBook is a fairly complex format and using and customizing the XSL style sheets available is not really straightforward. So by googling left and right and looking at other DocBook rendering frameworks that are in the open source, we tried to model similarities and sometimes just copied some of the ideas.

This DocBook Framework is literally standing on the shoulders of other projects, in particular:

- *The DocBook Format* by Norman Walsh; (C) 1999-2006 by Norman Walsh, OASIS and O'Reilly, especially all the documentation that is available from <http://www.docbook.org/>
- *The DocBook FAQ* maintained by Dave Pawson and available from <http://www.dpawson.co.uk/docbook/>. We wouldn't have survived without it.
- *DocBook XSL: The Complete Guide* by Bob Stayton. This is an invaluable reference to the DocBook style sheets. Find it online at <http://sagehill.net/> or buy the E-book.
- *The DocBook Project* located at <http://docbook.sourceforge.net/>. They maintain the XSL style sheets used to transform DocBook into other formats and also link to the docbook mailing list archives.
- *The Apache XML commons resolver* available from <http://xml.apache.org/commons/components/resolver/>
- *The XMLmind XML Editor* from XMLMind, available through <http://www.xmlmind.com/xmleditor/>. This cross-platform, pure Java editor not only runs well on Linux, Windows and MacOS but also offers DocBook WYSIWYG support and has a free version! And if you pay for it, you get the source code for it too.

Ideas on how to render elements, to arrange things and how to do more obscure things like title pages or use CSS to render HTML, we've taken (sometimes literally by cut and paste) from the following projects:

- *The Spring Framework documentation*. It hooked us on the idea that Velocity should have DocBook documentation, too. Their DocBook framework is really nice, however it proved to be 'not exactly what we were looking for' (see above). Spring is an example on how good documentation makes all the difference between a successful and popular project and 'the others'. Thanks a lot, Spring guys! Download Spring Framework from <http://www.springframework.org/>.
- *The "ant and docbook" styler suite* by Dawid Weiss, available from <http://www.cs.put.poznan.pl/dweiss/xml/projects/ant-docbook-styler/index.xml>. We stole his CSS style sheet almost verbatim. Thanks a lot, Dawid!
- *The Maven sdocbook plugin* by Siegfried Goeschl, Per Olesen and Carlos Sanchez, available at the SourceForge Maven plugin page at <http://maven-plugins.sourceforge.net/maven-sdocbook-plugin/>