

# How to use the JBoss DocBook system

A guide for content developers

Version: 1.0

---

# Table of Contents

Target Audience .....	iv
Preface .....	v
1. Introduction to DocBook processing .....	1
2. How to develop content for JBoss DocBook .....	3
2.1. Setup the directories .....	3
2.2. Author the content .....	5
2.3. Build the documents .....	5
3. Maintain the JBoss DocBook system .....	6

---

## List of Figures

1.1. The DocBook build process .....	2
2.1. The user guide DocBook in the AOP project .....	4
3.1. The docbook-support module .....	6

---

# Target Audience

All JBoss documentation content developers and style developers should read this document. You should also read this if you want to build JBoss DocBooks from the public CVS tree.

---

# Preface

This document introduces the new unified DocBook system for JBoss documentation in the public CVS tree. There are a couple of reasons why we did this.

- It provides a unified set of styles to make sure that all JBoss DocBooks have a consistent look and feel.
- The styles can be managed from a central location. We can design new styles or fix bugs without worrying about "deployment" issues.
- The libraries and build scripts can also be managed from a central location.
- The build process is simplified and standardized. Just follow the simple instructions in this guide to setup your docs directory and copy a very simple `build.xml` file.
- Last but not least, we can reduce the duplication of style sheets, XSLs and libraries in the system.

If you have questions, please feel free to contact Michael Yuan or Norman Richards for more information.

---

# 1

## Introduction to DocBook processing

DocBook is an XML format to write documents. It allows the author to focus on the content itself during the writing process instead of worrying about the presentation.

Using the standard DocBook tags, we can tag the content according to their syntax structure. The DocBook document is then processed against XSL style sheets. Each tagged element in the DocBook is transformed to a presentation element with the style (e.g., margin, font etc.) specified in the XSL. Using different XSL stylesheets, we can generate different output documents. For example, we can generate HTML and PDF outputs from a single DocBook source. We can also generate multiple versions of PDF (or HTML) files each with a different formatting style.

In the JBoss DocBook system, we provide XSL stylesheets to build HTML and PDF outputs from the DocBook source. The build process is illustrated in Figure 1.1.

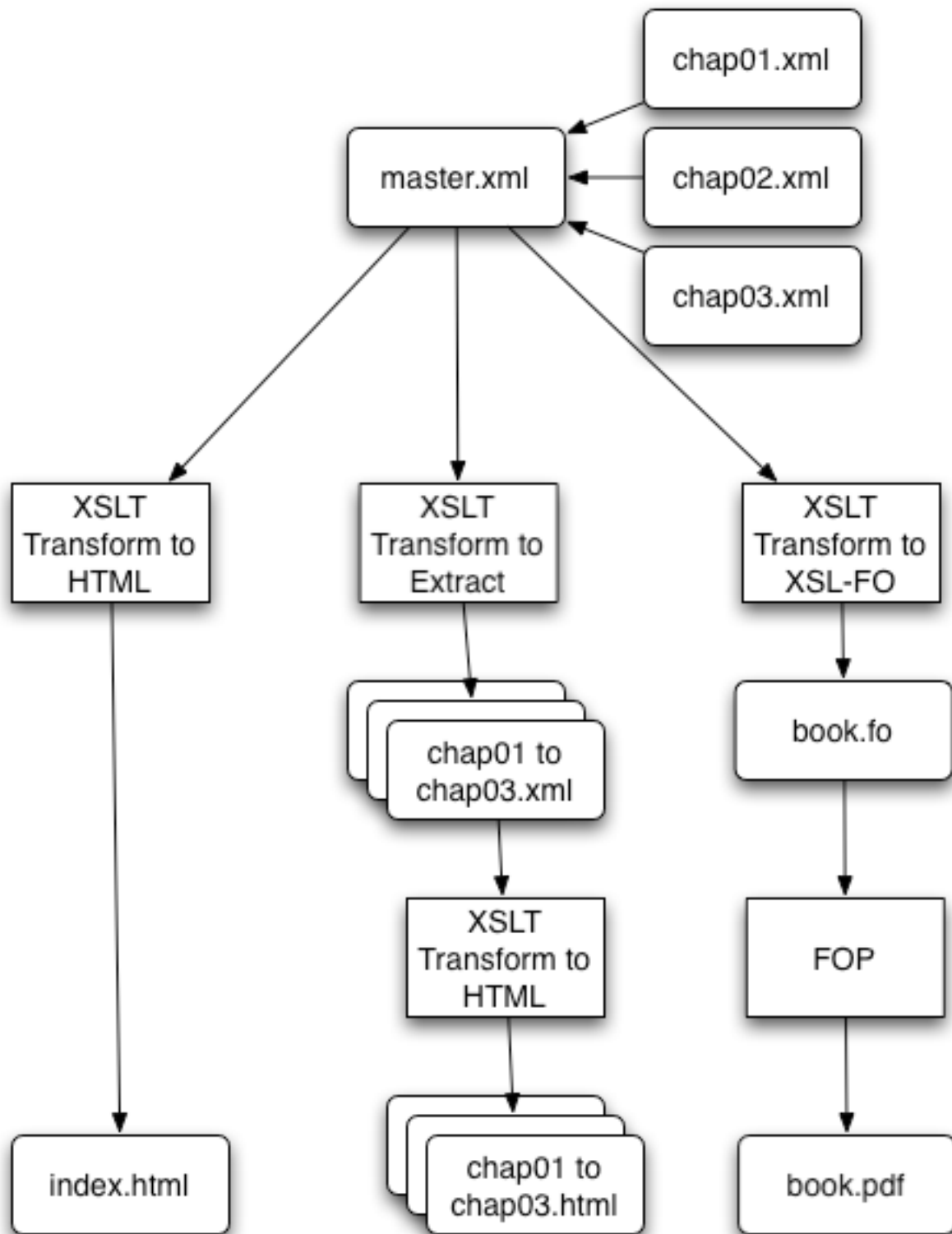


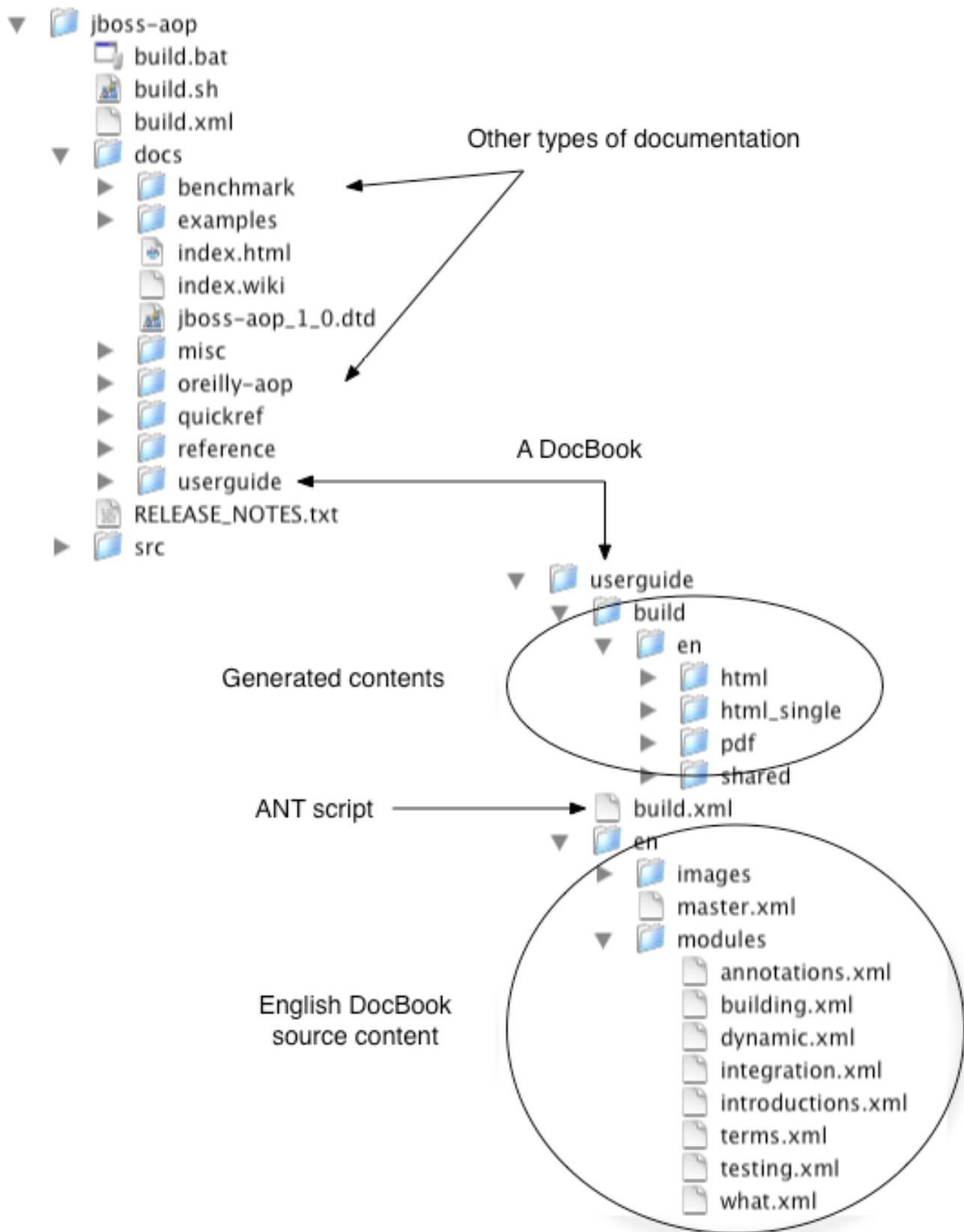
Figure 1.1. The DocBook build process

## How to develop content for JBoss DocBook

### 2.1. Setup the directories

Each top-level JBoss project (`projectname/`) in the public CVS stores its documentations in the `projectname/docs` directory. The `projectname/docs` directory can contain any number of subdirectories for API references, sample code, user guide etc. But each DocBook should be placed in its own directory directly under `projectname/docs`. For example, the user's guide DocBook for a project could be placed in `projectname/docs/userguide`. The easiest way to setup this DocBook directory is to copy the `docbook-support/docs/guide` directory to your target project and use it as a template. The `userguide` DocBook structure for the JBoss AOP project is shown in Figure 2.1.





**Figure 2.1. The user guide DocBook in the AOP project**

Inside the DocBook directory, there are typically several sub-directories, each corresponding to a specific language version of the document. The English version resides in the `en/` sub-directory. Inside each language directory, there are typically two sub-directories for contents:

- The `images/` directory stores images.
- The `modules/` directory stores DocBook text modules for each chapter.

## 2.2. Author the content

Now you can write your content in DocBook format. Make sure that the master file of your DocBook (i.e., the file that contains the `book` element) in the `master.xml` file directly under the language directory (see Figure 2.1). You can either put the entire content in `master.xml` or divide up the chapters and place them in the `modules/` directory. If you do the latter, you should reference the chapter files from the `master.xml` file via entity reference. Please see the `docbook-support/docs/guide/en/master.xml` file to see how it is done.

## 2.3. Build the documents

To build the deliverable documents, just run ANT against the `build.xml` file in the DocBook directory. The `build.xml` file is really simple and its content is shown below. It delegates most of the tasks to the `support.xml` file maintained by the `docbook-support` project.

```
<project name="Documentation" default="all.doc" basedir=". ">

  <property name="pdf.name" value="jboss-mybook.pdf" />
  <import file="../../../../docbook-support/support.xml" />

  <target name="all.doc" depends="clean">
    <antcall target="lang.all">
      <param name="lang" value="en"/>
    </antcall>
  </target>

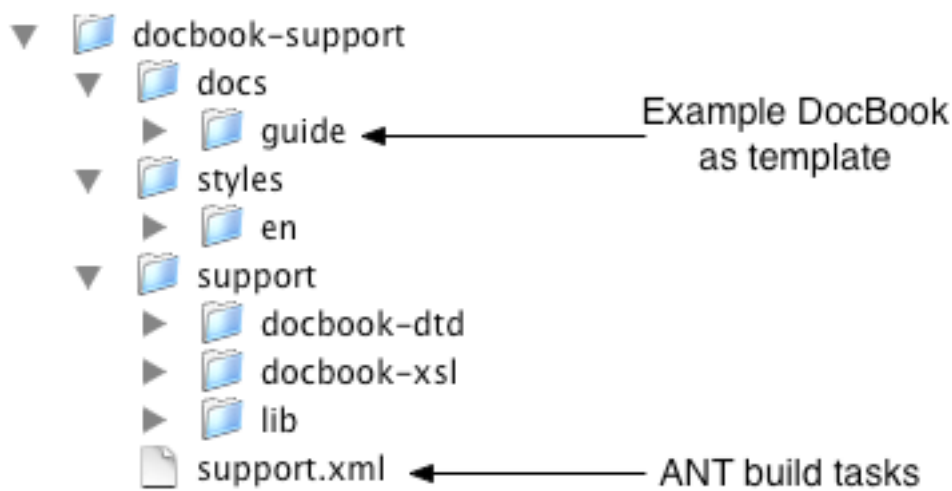
</project>
```

After the build is finished, you have three output documents for each language edition in the following places:

- The `build/en/html` directory contains the HTML version of the document. Each chapter is broken into a separate HTML file and they are linked by the `index.html` file.
- The `build/en/html_single` directory contains a big `index.html` file which holds the entire document.
- The `build/en/pdf` directory contains the PDF version of the document.

## Maintain the JBoss DocBook system

The structure of the `docbook-support` module is illustrated in Figure 3.1. The contents are as follows.



**Figure 3.1. The docbook-support module**

- The `support` directory contains Java libraries and executables for XML processors. It also contains standard DocBook XSL stylesheets. This is the place for "system" software.
- The `styles` directory contains DocBook styles we developed in-house for JBoss. Each language would have a separate set of styles.
- The `docs` directory contains this guide to serve as a template for other projects.
- The `support.xml` file contains all the necessary ANT tasks to build DocBooks. It is referenced from the `build.xml` file for each individual DocBook project.