ANT4DOCBOOK

User Manual

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 *APACHE 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.

undefined custom\_year

ant4docbook project

undefined custom\_date

undefined custom\_issue

Ant task for docbook

undefined custom\_ref

Benjamin

de Dardel

ANT4DOCBOOK Project

bdedardel[\_@t\_]users.sourceforge.net

Project leader

V0.1

2010/07/25

DocBook ant task : dbk to html generation

Benjamin

de Dardel

V0.2

2011/03/27

DocBook ant task : dbk to pdf generation

Benjamin

de Dardel

V0.3

2011/09/25

DocBook ant task : dbk to pdf customization with parameters

Benjamin

de Dardel

V0.4

2012/07/02

DocBook ant task : dbk to pdf images processing

Benjamin

de Dardel

V0.5

2012/12/16

dbk preprocessing with freemarker engine

Benjamin

de Dardel

V0.6

2014/01/05

dbk to epub processing using saxon Upgrade docbook xsl v1.78.1 Upgrade fop v1.1

Benjamin

de Dardel

V0.7

2014/04/06

docbook + plain text content (markdown)

Benjamin

de Dardel

V0.8

2016/01/03

docbook + graphics (Plantuml) process markdown as an input file

Benjamin

de Dardel

# Prerequisites

## Installation

**Apache ant**  
Download and install latest version of .  
Optionally set env variable

**Ant4docbook**  
Download latest version of on .  
Copy to , Copy to .

# Ant for docbook tasks

## DocBook ant task <dbk>

### Description

Ant4docbook is an ant task for processing docbook xml documents.  
see : *http://ant.apache.org/*  
see : *http://www.docbook.org/*

Ant4docbook transforms docbook files into other formats using :

* *DocBook XSL stylesheets* and XSL Formatting Objects,

see : *docbook to html*  
see : *docbook to pdf*  
see : *docbook to epub*

* Custom docx processor using *docx4java*  
  see :

Ant4docbook preprocess docbook files using *freemarker engine* to add variables and conditions on docbook files.  
see : *Templating with freemarker*

### Encoding

By default, all files are processed as UTF-8.

### DocBook task properties

DocBook task have parameters and inner elements to define behavior. Nevertheless, some global properties are useful to set some specific points.

## DocBook to html

### Description

Process docbook to html within ant script, using *DocBook XSL stylesheets*.

### Html task parameters

### Examples

*<project name="docbook-ant-task-to-html" default="docbook-to-html">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="docbook-to-html">*  
*<dbk file="docbook.xml" tofile="generated-html-file.html" />*  
*</target>*  
  
*<target name="docbook-to-html-with-many-parameters">*  
*<dbk file="docbook.xml" tofile="generated-html-file.html"*  
*parameters="html.properties" />*  
*</target>*  
  
*<target name="docbook-to-html-with-some-parameters">*  
*<dbk file="docbook.xml" tofile="generated-html-file.html">*  
*<parameter name="section.autolabel" value="1" />*  
*<parameter name="toc.section.depth" value="3" />*  
*</dbk>*  
*</target>*  
  
*</project>*

## DocBook to pdf

### Description

Process docbook to pdf within ant script, using *DocBook XSL stylesheets* and *XSL Formatting Objects*.

By default, ant4docbook embed apache fop to process pdf files.  
You may use your favorite fo processor as an external tool by setting property. For example :

*<project name="docbook-ant-task-to-pdf">*  
*<property name="ant4docbook.fop" value="/path/to/apache-fop/fop" />*  
*[...]*  
*</project>*

### Pdf task parameters

### Examples

*<project name="docbook-ant-task-to-pdf" default="docbook-to-pdf">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="docbook-to-pdf">*  
*<dbk file="docbook.xml" tofile="generated-pdf-file.pdf" />*  
*</target>*  
  
*<target name="docbook-to-pdf-with-many-parameters">*  
*<dbk file="docbook.xml" tofile="generated-pdf-file.pdf"*  
*parameters="pdf.properties" />*  
*</target>*  
  
*<target name="docbook-to-pdf-with-some-parameters">*  
*<dbk file="docbook.xml" tofile="generated-pdf-file.pdf">*  
*<parameter name="paper.type" value="A4" />*  
*</dbk>*  
*</target>*  
  
*</project>*

## DocBook to epub

### Description

Process docbook to epub within ant script, using *DocBook XSL stylesheets*.

### Epub task parameters

### Examples

*<project name="docbook-ant-task-to-epub" default="docbook-to-epub">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="docbook-to-epub">*  
*<dbk file="docbook.xml" tofile="generated-epub-file.epub" />*  
*</target>*  
  
*</project>*

## DocBook to docx

FUTURE RELEASE (experimental)

### Description

Existing docx stylesheets *DocBook XSL stylesheets* seems to be limited.  
The idea is to process docbook to docx within ant script, using a custom processor based on *docx4java*.

The beginning of the development is commited on the sourceforge repository. See dev/src/java/net/sourceforge/ant4docbook/processors/DocxProcessor.java. Some efforts are needed to continue the development and contributions are welcome.

### Docx task parameters

### Examples

*<project name="docbook-ant-task-to-docx" default="docbook-to-docx">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="docbook-to-docx">*  
*<dbk file="docbook.xml" tofile="generated-docx-file.docx" />*  
*</target>*  
  
*</project>*

# Docbook + Plain Text

## Markdown and DocBook

### Description

*Markdown* is a *lightweight markup languages*

* easy to read,
* easy to write,
* easy to convert, (see *ant + pandoc* )
* *easy to xinclude* in docbook documents. **This is (one of) the aim of ant4docbook.**

### Examples

docbook-markdown.xml

*<?xml version="1.0" encoding="utf-8"?>*  
*<article xmlns:xi="http://www.w3.org/2001/XInclude">*  
*<title>Test xinclude markdown</title>*  
*<xi:include href="docbook-markdown.md"/>*  
*</article>*

docbook-markdown.md

*A First Level Header*  
*====================*  
  
*A Second Level Header*  
*---------------------*  
  
*Now is the time for all good men to come to*  
*the aid of their country. This is just a*  
*regular paragraph.*  
  
*The quick brown fox jumped over the lazy*  
*dog's back.*  
  
*### Header 3*  
  
*> This is a blockquote.*  
*>*  
*> This is the second paragraph in the blockquote.*  
*>*  
*> ## This is an H2 in a blockquote*

build.xml

*<project name="docbook-markdown" default="docbook-and-markdown">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<!-- xinclude markdown -->*  
*<target name="xinclude-markdown">*  
*<dbk file="docbook-markdown.xml" tofile="docbook-markdown.pdf" />*  
*</target>*  
  
*<!-- or process markdown -->*  
*<target name="process-markdown">*  
*<dbk file="markdown.md" tofile="markdown.html" />*  
*</target>*  
  
*</project>*

see: *docbook-markdown.pdf*

# Docbook + Graphics

## DocBook and plantuml

### Description

*Plantuml* is a modeling language to create *UML* diagrams (use case, class, sequence, activity, component, state, ...).

Ant4docbook aims to:

* easy convert plantuml to svg or png, using *ant + plantuml*.
* easy xinclude plantuml in docbook documents.

### Installation

Please install (linux debian):  
$ apt-get install graphviz

### Examples

docbook-plantuml.xml

*<?xml version="1.0" encoding="utf-8"?>*  
*<article xmlns:xi="http://www.w3.org/2001/XInclude">*  
*<title>Test xinclude plantuml</title>*  
*<xi:include href="diagram.plantuml"/>*  
*</article>*

diagram.plantuml

*\@startuml*  
*Alice -> Bob: Authentication Request*  
*Bob --> Alice: Authentication Response*  
  
*Alice -> Bob: Another authentication Request*  
*Alice <-- Bob: another authentication Response*  
*\@enduml*

build.xml

*<project name="docbook-plantuml" default="docbook-and-plantuml">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="docbook-plantuml">*  
*<dbk file="docbook-plantuml.xml" tofile="docbook-plantuml.pdf" />*  
*</target>*  
  
*</project>*

see: *docbook-plantuml.pdf*

# DocBook templating

## Preprocessing with freemarker

### Description

Ant4docbook preprocess docbook files as *freemarker templates* , to add variables and conditions if needed.

### Preprocessing parameters

### Examples

DocBook file with freemarker instructions :

*<?xml version="1.0" encoding="utf-8"?>*  
*<article xml:lang="en">*  
  
*<title>$ {custom\_title}</title>*  
  
*<section id="section1">*  
*<title>Section level 1</title>*  
*<para>*  
*write your documentation with some conditions*  
*[# if (output\_format\_is\_pdf)??]*  
*<xref linkend="appendix-xxx" />*  
*[# else]*  
*<ulink url="appendix-xxx.html">appendix-xxx</ulink>*  
*[/ #if]*  
*</para>*  
*</section>*  
  
*</article>*

Ant script with freemarker parameters :

*<project name="ant-project" default="docbbook-preprocessing">*  
  
*<target name="docbbook-preprocessing">*  
*<!-- define <dbk> ant task here, with preprocessor parameter set to true -->*  
*<dbk file="docbook.xml" tofile="file.html" preprocessor="true">*  
*<parameter name="custom\_title" value="My Custom Title" />*  
*</dbk>*  
*<dbk file="docbook.xml" tofile="file.pdf" preprocessor="true">*  
*<parameter name="custom\_title" value="My Custom Title" />*  
*<parameter name="output\_format\_is\_pdf" value="true" />*  
*</dbk>*  
*</target>*  
  
*</project>*

# DocBook processing

## DocBook debugging

Ant4docbook let you process docbook step by step to debug errors :

1. docbook to xml, to process xinclude
2. docbook to fo
3. fo to pdf

Example :

*<project name="docbook-debugging" default="debug-pdf">*  
  
*<taskdef name="dbk"*  
*classname="net.sourceforge.ant4docbook.taskdefs.DocbookTask"*  
*classpath="PATH/TO/ant4docbook.jar"/>*  
  
*<target name="validate-dtd-or-not">*  
*<property name="ant4docbook.dtd.validating" value="true\*|false"/>*  
*<dbk file="docbook.xml" tofile="docbook.xinclude.xml" />*  
*</target>*  
  
*<target name="debug-pdf">*  
*<dbk file="docbook.xml" tofile="docbook.xinclude.xml" />*  
*<dbk file="docbook.xinclude.xml" tofile="docbook.fo" />*  
*<dbk file="docbook.fo" tofile="docbook.pdf" />*  
*</target>*  
  
*<target name="debug-html">*  
*<dbk file="docbook.xml" tofile="docbook.xinclude.xml" />*  
*<dbk file="docbook.xinclude.xml" tofile="docbook.html" />*  
*</target>*  
  
*<target name="debug-with-your-xsl-stylesheet">*  
*<dbk file="docbook.xml" tofile="custom-docbook.xml"*  
*stylesheet="/path/to/your/xsl/stylesheet" />*  
*</target>*  
  
*</project>*