

# ANT4DOCBOOK

## User Manual

### Ant task for docbook



**V0.8.0**

**Copyright © 2016 ant4docbook project**

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 \[http://www.apache.org/licenses/LICENSE-2.0\]](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.

## Authors

Benjamin de Dardel, ANT4DOCBOOK Project <bdedardel[\_@t\_]users.sourceforge.net>

### Revision History

Revision V0.1	2010/07/25	Benjamin de Dardel
DocBook ant task : dbk to html generation		
Revision V0.2	2011/03/27	Benjamin de Dardel
DocBook ant task : dbk to pdf generation		
Revision V0.3	2011/09/25	Benjamin de Dardel
DocBook ant task : dbk to pdf customization with parameters		
Revision V0.4	2012/07/02	Benjamin de Dardel
DocBook ant task : dbk to pdf images processing		
Revision V0.5	2012/12/16	Benjamin de Dardel
dbk preprocessing with freemarker engine		
Revision V0.6	2014/01/05	Benjamin de Dardel
dbk to epub processing using saxon		
Upgrade docbook xsl v1.78.1		
Upgrade fop v1.1		
Revision V0.7	2014/04/06	Benjamin de Dardel
docbook + plain text content (markdown)		
Revision V0.8	2016/01/03	Benjamin de Dardel
docbook + graphics (Plantuml)		
process markdown as an input file		

## Table of Contents

1. Prerequisites .....	1
1.1. Installation .....	1
2. Ant for docbook tasks .....	2
2.1. DocBook ant task <dbk> .....	2
Description .....	2
Encoding .....	3
DocBook task properties .....	3
2.2. DocBook to html .....	4
Description .....	4
Html task parameters .....	4
Examples .....	4
2.3. DocBook to pdf .....	6
Description .....	6
Pdf task parameters .....	6
Examples .....	7
2.4. DocBook to epub .....	8
Description .....	8
Epub task parameters .....	8
Examples .....	8
2.5. DocBook to docx .....	10
Description .....	10
Docx task parameters .....	10
Examples .....	11
3. Docbook + Plain Text .....	12
3.1. Markdown and DocBook .....	12
Description .....	12
Examples .....	12
4. Docbook + Graphics .....	14
4.1. DocBook and plantuml .....	14
Description .....	14
Installation .....	14
Examples .....	14
5. DocBook templating .....	16
5.1. Preprocessing with freemarker .....	16
Description .....	16
Preprocessing parameters .....	16
Examples .....	16
6. DocBook processing .....	18
6.1. DocBook debugging .....	18
Pdf parameters .....	19

# 1. Prerequisites

## 1.1 Installation

### 1. Apache ant

Download and install latest version of **apache ant** [<http://ant.apache.org>] .

Optionally set env variable

```
# Example for linux ANT_HOME=/path/to/apache-ant PATH=$ANT_HOME/  
bin:$PATH
```

### 2. Ant4docbook

Download latest version of **ant4docbook** on sourceforge [<http://sourceforge.net/projects/ant4docbook/files/ant4docbook/>] .

Copy ant4x.jar to \$ANT\_HOME/lib,

Copy ant4docbook.jar to \$ANT\_HOME/lib.

## 2. Ant for docbook tasks

### 2.1 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 :

see : Section 2.2, "DocBook to html"

see : Section 2.3, "DocBook to pdf"

see : Section 2.4, "DocBook to epub"

- DocBook XSL stylesheets [<https://sourceforge.net/projects/docbook/>] and XSL Formatting Objects,
- Custom docx processor using docx4java [<http://www.docx4java.org>]

see : Section 2.5, "DocBook to docx"

Ant4docbook preprocess docbook files using freemarker engine [<http://freemarker.sourceforge.net/>] to add variables and conditions on docbook files.

see : Section 5.1, "Preprocessing with freemarker"

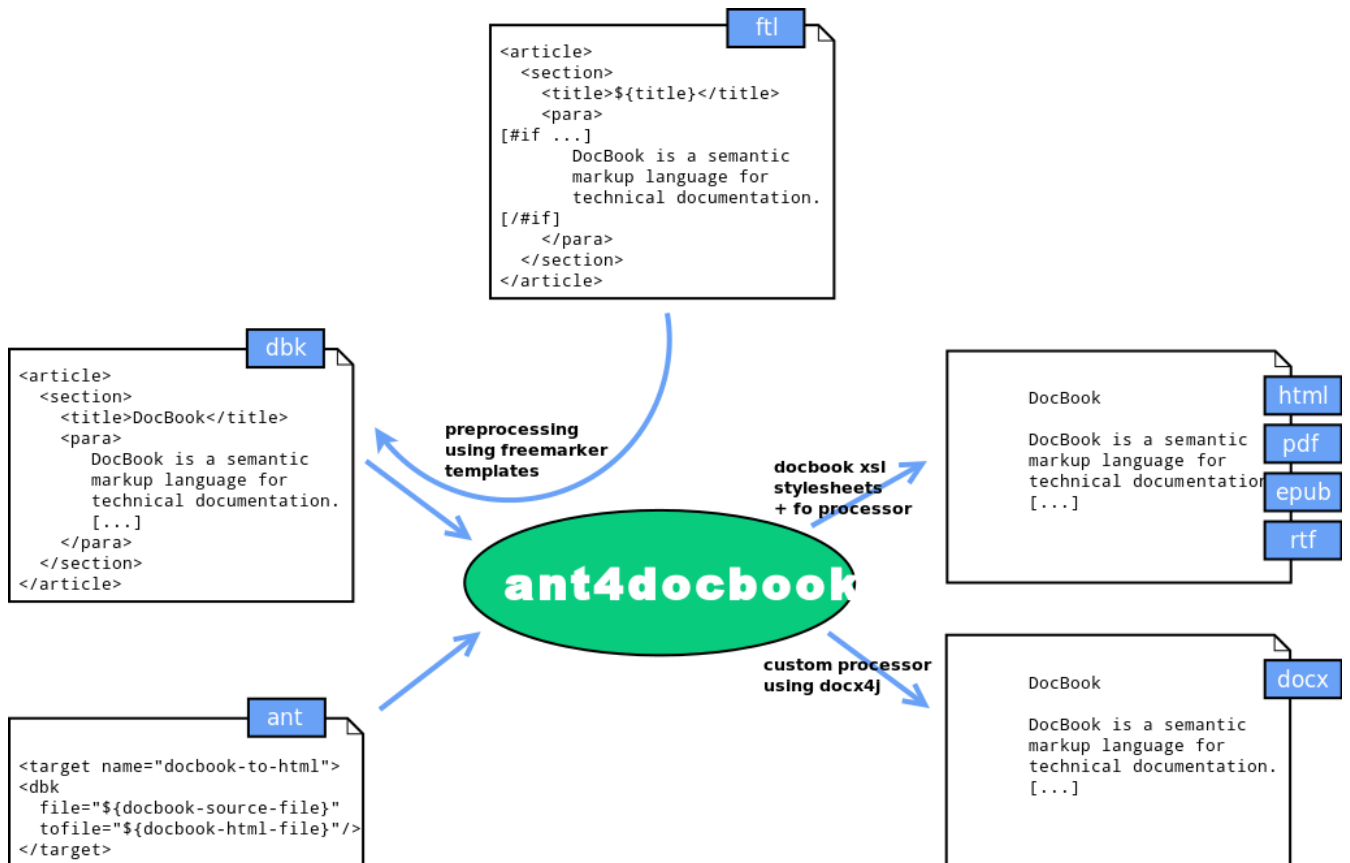


Figure 2.1. DocBook processing with ant

## 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.

Table 2.1. DocBook task properties

Property	Description	Value	Required
ant4docbook.keepTempFiles	keep temp files (for debugging)	<property name="ant4docbook.keepTempFiles" value="true false*"/>	no
ant4docbook.dtd.validating	use dtd validator	<property name="ant4docbook.dtd.validating" value="true false*"/>	no
ant4docbook.fop	external fo processor	/path/to/fo/processor	no

## 2.2 DocBook to html

### Description

Process docbook to html within ant script, using DocBook XSL stylesheets [<http://docbook.sourceforge.net/>].

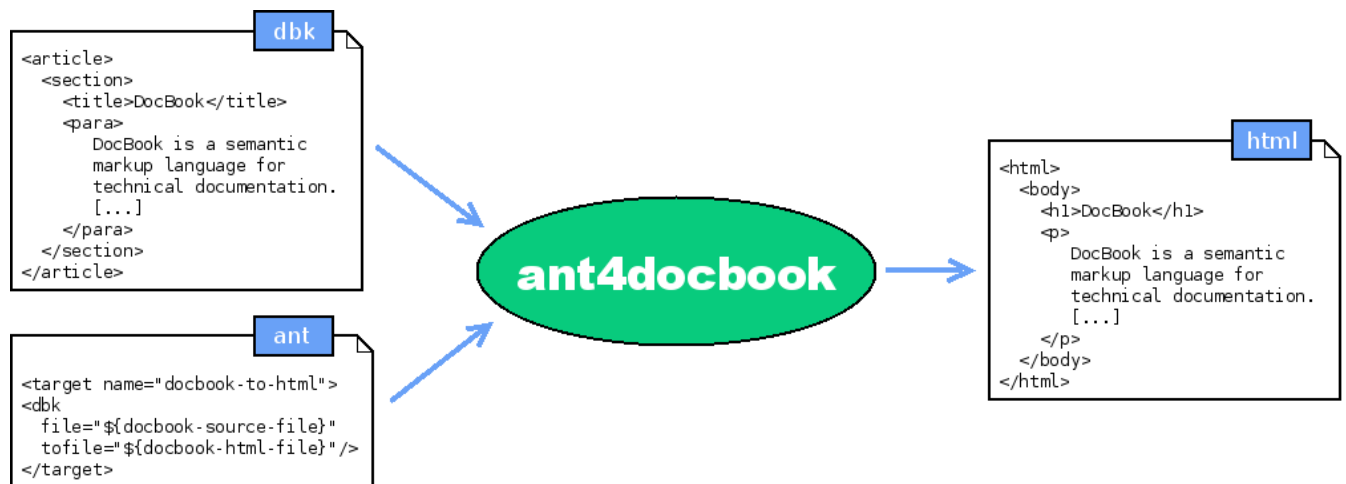


Figure 2.2. DocBook to html processing

### Html task parameters

Table 2.2. Html task parameters

Attribute	Description	Value	Required
file	docbook file		yes
tofile	output file	<file name>.html	yes
stylesheet	custom stylesheet	<my-stylesheet>.xsl	no
css	custom css	<my-css>.css	no
parameters	output parameters		no
Element	Description	Value	Required
parameter	output parameters		no

### 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>
```



## 2.3 DocBook to pdf

### Description

Process docbook to pdf within ant script, using DocBook XSL stylesheets [<http://docbook.sourceforge.net/>] and XSL Formatting Objects [[http://en.wikipedia.org/wiki/XSL\\_Formatting\\_Objects](http://en.wikipedia.org/wiki/XSL_Formatting_Objects)].

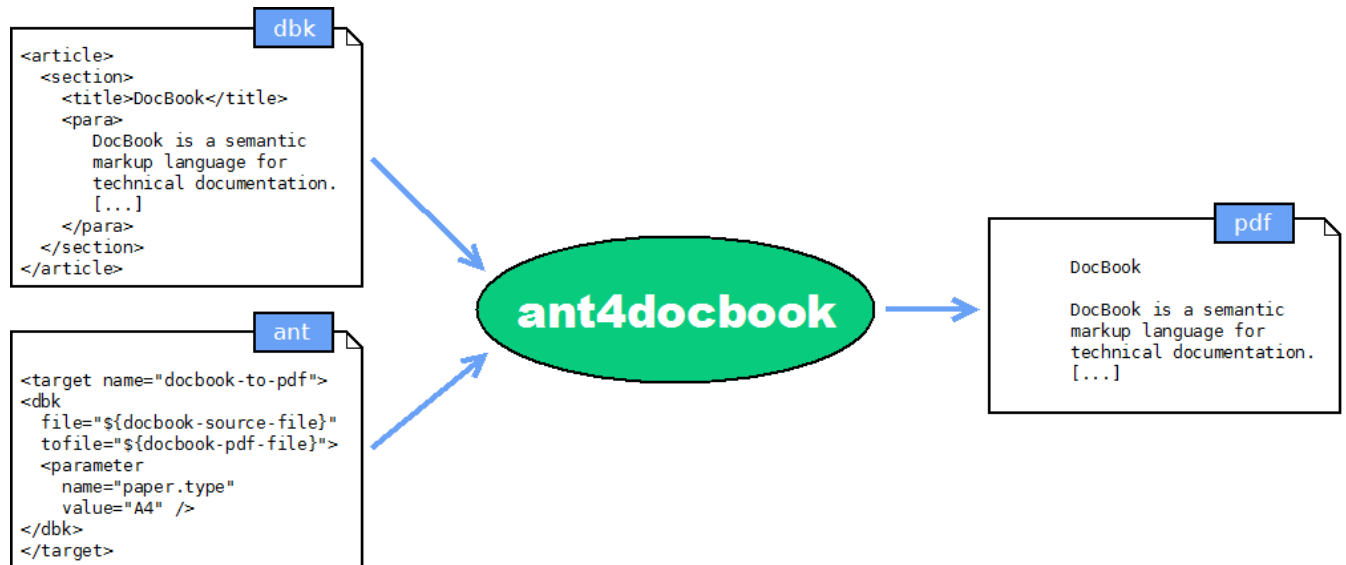


Figure 2.3. DocBook to pdf processing

By default, ant4docbook embed apache fop to process pdf files.

You may use your favorite fo processor as an external tool by setting `ant4docbook.fop` property. For example :

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

### Pdf task parameters

Table 2.3. Pdf task parameters

Attribute	Description	Value	Required
file	docbook file		yes
tofile	output file	<file name>.pdf	yes

Attribute	Description	Value	Required
parameters	output parameters	see <i>Pdf parameters</i>	no
Element	Description	Value	Required
parameter	output parameters	see <i>Pdf parameters</i>	no

## 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>
```

## 2.4 DocBook to epub

### Description

Process docbook to epub within ant script, using DocBook XSL stylesheets [<http://docbook.sourceforge.net/>].

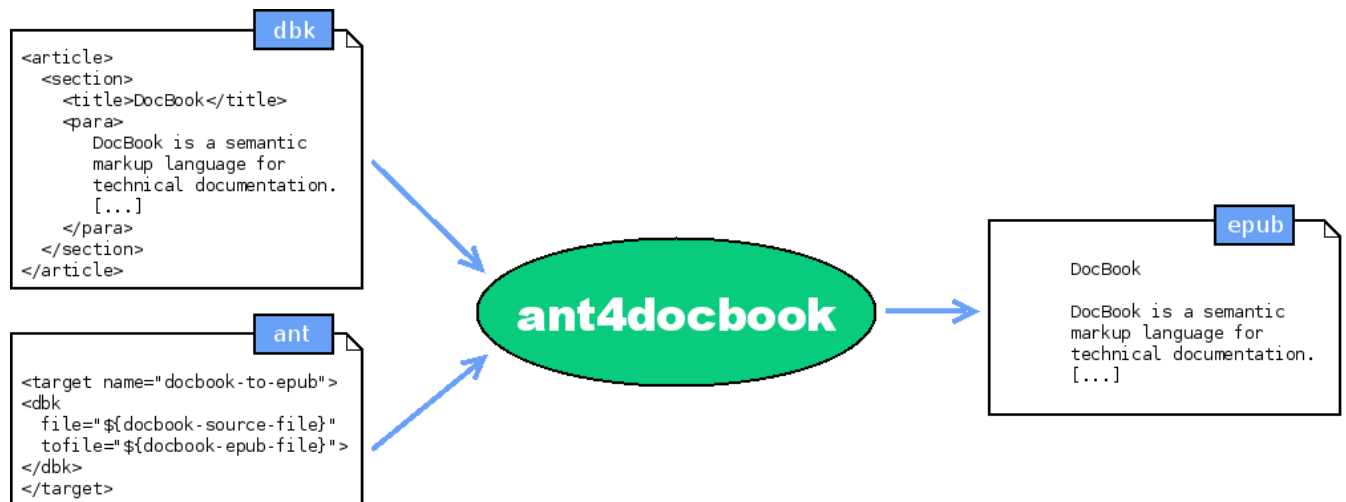


Figure 2.4. DocBook to epub processing

### Epub task parameters

Table 2.4. Epub task parameters

Attribute	Description	Value	Required
file	docbook file		yes
tofile	output file	<file name>.epub	yes
style	css file	/path/to/custom/css	yes

### 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>
```

```
</target>  
  
</project>
```

## 2.5 DocBook to docx

### Note

FUTURE RELEASE (experimental)

## Description

Existing docx stylesheets DocBook XSL stylesheets [<http://docbook.sourceforge.net/>] seems to be limited.

The idea is to process docbook to docx within ant script, using a custom processor based on docx4java [<http://www.docx4java.org/>].

The beginning of the development is committed on the sourceforge repository. See [dev/src/java/net/sourceforge/ant4docbook/processors/DocxProcessor.java](http://dev/src/java/net/sourceforge/ant4docbook/processors/DocxProcessor.java). Some efforts are needed to continue the development and contributions are welcome.

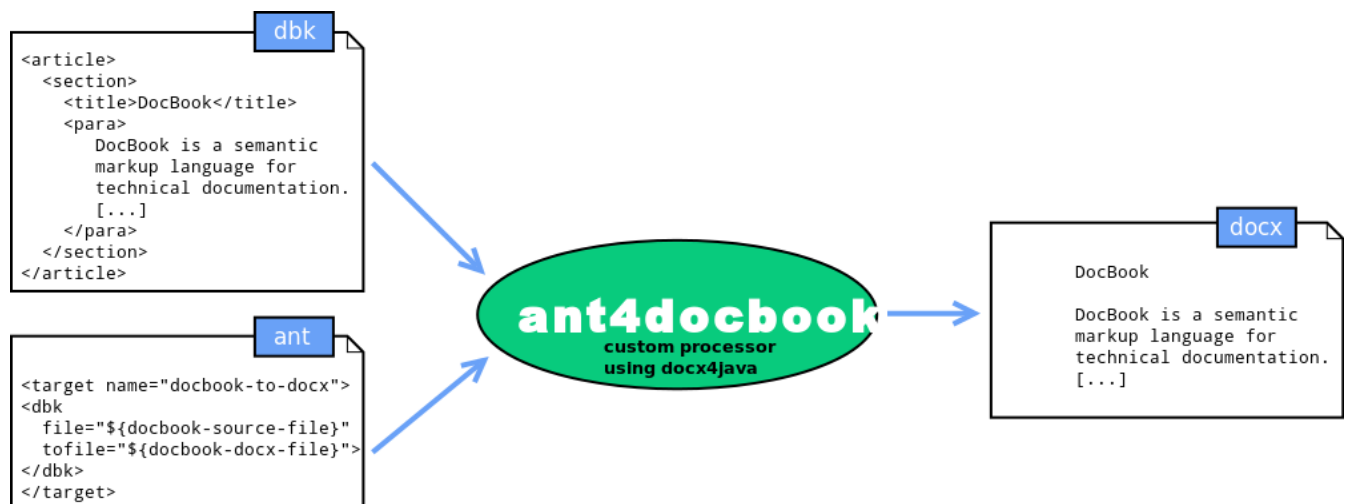


Figure 2.5. DocBook to docx processing

## Docx task parameters

Table 2.5. Docx task parameters

Attribute	Description	Value	Required
file	docbook file		yes
tofile	output file	<file name>.docx	yes
Element	Description	Value	Required
parameter	template wordml	<parameter name="wordml.template" value="path:to/docbook-	no

Element	Description	Value	Required
		xsl/roundtrip/ template.xml"/>	

## 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>
```

## 3. Docbook + Plain Text

### 3.1 Markdown and DocBook

#### Description

Markdown [<http://en.wikipedia.org/wiki/Markdown>] is a lightweight markup languages [[http://en.wikipedia.org/wiki/Category:Lightweight\\_markup\\_languages](http://en.wikipedia.org/wiki/Category:Lightweight_markup_languages)]

- easy to read,
- easy to write,
- easy to convert, (see ant + pandoc [<http://ant4x.sourceforge.net/ant-pandoc.html>])
- *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](#) [[samples/docbook-markdown.pdf](#)]



## 4. Docbook + Graphics

### 4.1 DocBook and plantuml

#### Description

Plantuml [<http://plantuml.sourceforge.net/>] is a modeling language to create UML [[http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)] diagrams (use case, class, sequence, activity, component, state, ...).

Ant4docbook aims to:

- easy convert plantuml to svg or png, using ant + plantuml [<http://ant4x.sourceforge.net/ant-plantuml.html>].
- 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](#) [[samples/docbook-plantuml.pdf](#)]

## 5. DocBook templating

### 5.1 Preprocessing with freemarker

#### Description

Ant4docbook preprocess docbook files as freemarker templates [http://freemarker.sourceforge.net/], to add variables and conditions if needed.

#### Preprocessing parameters

Table 5.1. Preprocessing parameters

Attribute	Description	Value	Required
file	docbook file		yes
tofile	output file	<file name>.pdf	yes
preprocessor		true false	yes
Element	Description	Value	Required
parameter	freemarker parameters	/\ DO NOT USE DOT [.] IN VARIABLE NAMES	no

#### 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>
```

## 6. DocBook processing

### 6.1 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>
```

## Pdf parameters

```
# see : http://docbook.sourceforge.net/release/xsl/1.76.1/doc/fo/general.html

# The height of the physical page
page.height=

# Specify the physical size of the long edge of the page
page.height.portrait=

# The bottom margin of the page
page.margin.bottom=

# The inner page margin
page.margin.inner=

# The outer page margin
page.margin.outer=

# The top margin of the page
page.margin.top=

# Select the page orientation
page.orientation=

# The width of the physical page
page.width=

# Specify the physical size of the short edge of the page
page.width.portrait=

# Select the paper type
paper.type=A4

# Is the document to be printed double sided?
double.sided=

# The bottom margin of the body text
body.margin.bottom=

# To specify the size of the top margin of a page
body.margin.top=
```

```
# The start-indent for the body text
body.start.indent=

# The end-indent for the body text
body.end.indent=

# Specify the default text alignment
alignment=

# Specify hyphenation behavior
hyphenate=

# Specify the line-height property
line-height=

# Number of columns on back matter pages
column.count.back=

# Number of columns on body pages
column.count.body=

# Number of columns on front matter pages
column.count.front=

# Number of columns on index pages
column.count.index=

# Number of columns on a 'List-of-Titles' page
column.count.lot=

# Number of columns on a title page
column.count.titlepage=

# Gap between columns in back matter
column.gap.back=

# Gap between columns in the body
column.gap.body=

# Gap between columns in the front matter
column.gap.front=

# Gap between columns in the index
column.gap.index
```

```
# Gap between columns on a 'List-of-Titles' page
column.gap.lot=

# Gap between columns on title pages
column.gap.titlepage=

# Specifies the height of the footer.
region.after.extent

# Specifies the height of the header
region.before.extent

# Default units for an unqualified dimension
default.units=

# What space do you want between normal paragraphs
normal.para.spacing=

# Specifies the default point size for body text
body.font.master=

# Specifies the default font size for body text
body.font.size=

# The font size for footnotes
footnote.font.size=

# Adjust the left margin for titles
title.margin.left=

# Select draft mode
draft.mode=

# The URI of the image to be used for draft watermarks
draft.watermark.image=

# Put headers on blank pages?
headers.on.blank.pages=

# Put footers on blank pages?
footers.on.blank.pages=

# Rule under headers?
header.rule=
```



```
# Rule over footers?
footer.rule=

# Specify relative widths of header areas
header.column.widths=

# Specify relative widths of footer areas
footer.column.widths=

# Apply properties to the header layout table
header.table.properties=

# Specify the minimum height of the table containing the running page headers
header.table.height=

# Apply properties to the footer layout table
footer.table.properties=

# Specify the minimum height of the table containing the running page footers
footer.table.height=

# Properties of page header content
header.content.properties=

# Properties of page footer content
footer.content.properties=

# Control depth of sections shown in running headers or footers
marker.section.level=
```