

Release Notes

XMLMill v 1.35

1.0

xmlmill

© 2000-2002 Pecunia Data Systems bvba. All rights reserved.

NOTICE: All information contained herein is the property of Pecunia Data Systems bvba.

This publication and the information herein are furnished AS IS, are subject to change without notice, and should not be construed as a commitment by Pecunia Data Systems bvba. Pecunia Data Systems bvba assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied, or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes, and noninfringement of third-party rights. (First printing, January 2003)

Table of Contents

1. Disclaimer.....	8
2. Preface.....	9
3. Prerequisites.....	10
4. Upgrading from previous versions.....	11
4.1. Upgrading from 1.32.....	11
4.1.1. text-wrapping is per default off.....	11
4.1.2. Keywords are replaced by <tags>.....	11
4.2. Upgrading from earlier versions.....	11
5. Download.....	13
5.1. Distributions.....	13
5.2. Extra downloads.....	13
5.3. Distribution's structure.....	13
5.4. Signed jar's package structure.....	14
6. XMLMill Interactive.....	15
6.1. XMLMill in a browser.....	15
6.2. XMLMill as application.....	15
6.3. Renewed interface.....	16
6.3.1. Menu options.....	16
6.3.1.1. File Exit.....	17
6.3.1.2. Log.....	17
6.3.1.2.1. Comments.....	17
6.3.1.2.2. Warnings.....	17
6.3.1.2.3. Errors.....	17
6.3.1.2.4. Abort on Warning.....	17
6.3.1.2.5. Abort on Error.....	17
6.3.1.2.6. PrintStackTrace.....	17
6.3.1.3. Help About.....	18
6.4. Buttons.....	18
6.4.1. Open.....	18
6.4.2. Output.....	18
6.4.3. Log.....	18
6.4.4. XSL.....	18
6.4.5. Convert.....	18
6.4.6. Reset.....	19
6.4.7. XSLT Transformation only.....	19
7. XMLMill batch.....	20
7.1. Support for 'headless' environments.....	20
7.2. New parameters to support new functionality.....	21
8. XMLMill in Servlets.....	22
8.1. Class: PDXTransform.....	22
8.2. Class: PDXTransformErrorHandler.....	22
8.2.1. public PDXTransformErrorHandler().....	22
8.2.2. public PDXTransformErrorHandler(int flags).....	22
8.2.3. void error (SAXParseException e).....	22

8.2.4. void fatalerror (SAXParseException e).....	23
8.2.5. void warning (SAXParseException e).....	23

9. XMLMill in applications..... 24

10. Images..... 25

10.1. Image processing.....	25
10.2. New image types:.....	25
10.2.1. PCX - PC Paintbrush.....	25
10.2.2. BMP - Windows Bitmap.....	26
10.2.3. DWG - AutoCAD R13/R14/R2000 DWG File.....	26

11. New tags..... 27

11.1. <navigation>.....	27
11.1.1. Attribute: number-format.....	27
11.1.2. Attribute: outlines.....	27
11.1.3. Attribute: table-of-contents.....	27
11.2. <viewerpreferences>.....	28
11.2.1. Attribute: centerwindow.....	28
11.2.2. Attribute: fitwindow.....	28
11.2.3. Attribute: hidemenubar.....	28
11.2.4. Attribute: hidetoolbar.....	29
11.2.5. Attribute: hidewindowui.....	29
11.3. <table-of-contents>.....	29
11.3.1. Attribute: show-levels.....	29
11.3.2. Attribute: show-pagenumber.....	29
11.3.3. Attribute: right-align-page-number.....	30
11.3.4. Attribute: leader-pattern.....	30
11.3.5. <toc-header>.....	30
11.3.6. <toc-style>.....	30
11.3.7. Attribute: level.....	30
11.4. <list-numbers>.....	30
11.4.1. Attribute: number-format.....	31
11.4.2. Attribute: label-indent.....	31
11.4.3. Attribute: text-indent.....	31
11.4.4. Attribute: start-at.....	31
11.4.5. <list-label-format>.....	31
11.4.6. <list-item-text>.....	31
11.5. <list-bullets>.....	31
11.5.1. Attribute: label-indent.....	32
11.5.2. Attribute: text-indent.....	32
11.5.3. <list-label>.....	32
11.5.4. <list-item-text>.....	32
11.6. <page-number>.....	32
11.7. <page-total>.....	32
11.8. <file>.....	33
11.9. <date>.....	33
11.10. <time>.....	33
11.11. <datetime>.....	34
11.11.1. Attribute: format.....	34
11.12. <link>.....	35
11.12.1. Attribute: internal-destination.....	35
11.12.2. Attribute: external-destination.....	35

12. Existing tags.	36
12.1. <documents>	36
12.1.1. Attribute: whitespace-collapse.	36
12.1.2. Attribute: space-collapse.	37
12.1.3. Attribute: space-treatment.	38
12.1.4. Attribute: linefeed-treatment.	38
12.2. <document>	39
12.2.1. Attribute: page-mode.	39
12.2.2. Attribute: page-layout.	39
12.3. <info>	40
12.4. <p>	40
12.4.1. Attribute: hanging-indent.	40
12.4.2. Attribute: auto-numbering.	40
12.4.3. Attribute: outline-level.	40
12.4.4. Attribute: show-in-outlines.	40
12.4.5. Attribute: show-in-toc.	41
12.4.6. Attribute: link-id.	41
12.4.7. Attribute: linefeed-treatment.	41
12.4.8. Attribute: space-collapse.	41
12.4.9. Attribute: space-treatment.	41
12.4.10. Attribute: whitespace-collapse.	41
12.4.11. Attribute: script-mode.	41
12.5. <textbox>	41
12.6. <image>	41
12.6.1. Attribute: alternate.	41
12.7. <inline>	42
12.7.1. Attribute: linefeed-treatment.	42
12.7.2. Attribute: space-collapse.	42
12.7.3. Attribute: space-treatment.	42
12.7.4. Attribute: whitespace-collapse.	42
12.7.5. Attribute: script-mode.	42
12.8. <header>	42
12.8.1. Attribute: border-bottom.	42
12.8.2. Attribute: border-bottom-color.	42
12.8.3. Attribute: border-bottom-style.	42
12.8.4. Attribute: border-bottom-width.	43
12.8.5. Attribute: linefeed-treatment.	43
12.8.6. Attribute: space-collapse.	43
12.8.7. Attribute: space-treatment.	43
12.8.8. Attribute: whitespace-collapse.	43
12.9. <footer>	43
12.9.1. Attribute: border-top.	43
12.9.2. Attribute: border-top-color.	44
12.9.3. Attribute: border-top-style.	44
12.9.4. Attribute: border-top-width.	44
12.9.5. Attribute: linefeed-treatment.	44
12.9.6. Attribute: space-collapse.	44
12.9.7. Attribute: space-treatment.	45
12.9.8. Attribute: whitespace-collapse.	45
12.9.9. Attribute: script-mode.	45
12.10. <thead>	45
12.10.1. Attribute: hanging-indent.	45
12.11. <rows>	45
12.11.1. Attribute: hanging-indent.	45

12.12. <row>.....	45
12.12.1. Attribute: hanging-indent.....	45
12.13. <field>.....	45
12.13.1. Attribute: auto-numbering.....	45
12.13.2. Attribute: link-id.....	45
12.13.3. Attribute: outline-level.....	45
12.13.4. Attribute: show-in-outlines.....	45
12.13.5. Attribute: show-in-toc.....	45
13. Known Bugs.....	46

xmlmill

xmlmill

1. Disclaimer

This software redistributes parts of The Apache Software Foundation binary code, hence the disclaimer below:

The Apache Software License, Version 1.1

- * Copyright (c) 1999 The Apache Software Foundation. All rights reserved.
- *
- * Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- *
- * 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- *
- * 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- *
- * 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:
 - * "This product includes software developed by the
 - * Apache Software Foundation (<http://www.apache.org/>)."
 - * Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
- *
- * 4. The names "Xalan" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
- *
- * 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.
- *
- * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
- * =====
- *
- * This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, Lotus Development Corporation., <http://www.lotus.com>. For more information on the Apache Software Foundation, please see [<http://www.apache.org/>](http://www.apache.org/).
- *

2. Preface

The XMLMill application is intended for software developers or end-users who want to generate .pdf documents from xml and/or xsl.

This release notes document describes all new (or enhanced) functionalities of version 1.35 of XMLMill.

For a full overview of all the tags / attributes please consult the xmlmill.dtd.

More information can be found at the [website](#).

If you have questions, please do not hesitate to send a mail to support@xmlmill.com.

Remark:

☞ *This document is completely generated with XMLMill 1.35 using **rnotes135.xml** and **rnotes.xsl**. These files can be found in the directory **samples/docs/rnotes** in the download.*

xmlmill

3. Prerequisites

XMLMill v1.35 is developed in Java and needs as such a 'Java enabled' environment. Following need to be installed:

- ◆ A Java 2 compatible SDK/JRE, version 1.3.1_01 or higher.
- ◆ For an application server: supporting Servlets.
- ◆ A valid version of [Xalan-Java version 2](#).
- ◆ A valid version of [Xerces2 Java Parser](#).

XMLMill v1.35 can run in following environments:

- ◆ In a browser using an applet (see the xmlmill.html file in the download).
- ◆ As a stand-alone Java application (see the gui.cmd or gui.sh in the download).
- ◆ In batch, running from the command-line (support for 'headless' (no X11) Linux/Unix servers (see the batch.cmd or batch.sh in the download).
- ◆ In an Java Application server (see the /testservlet directory in the download for more information).

xmlmill

4. Upgrading from previous versions

4.1. Upgrading from 1.32

4.1.1. text-wrapping is per default off

There is one important change regarding the previous version:

*The **text-wrapping** attribute is now per default off (in previous releases in was **on**).*

It was changed because a lot of users were confused when using XMLMill for the first time (seeing objects on pages 'flying' around all over the page instead of below each-other). Therefore as of this version text-wrapping is per default off, not on).

Please check your xml/xsl files and adapt following tags:

- ◆ **<line>**
- ◆ **<image>**
- ◆ **<box>**
- ◆ **<textbox>**
- ◆ **<table>**
- ◆ **<p>**

as follows:

- ◆ Check if a tag should be wrapping other elements around it.
- ◆ If so, check if the **text-wrapping="on"** is mentioned as an attribute.
- ◆ If not, please add the **text-wrapping="on"** attribute to the tag.

4.1.2. Keywords are replaced by <tags>

Following keywords were replaced by a tag:

- [Page] replaced by **<page-number>**
- [Pages] replaced by **<page-total>**
- [File] replaced by **<file>**
- [Time] replaced by **<time>**
- [Datetime] replaced by **<datetime>**

Please update your .xml/.xsl files accordingly.

Please do not forget to replace the xmlmill.dtd of the previous version with the xmlmill.dtd found in the download.

4.2. Upgrading from earlier versions

The best procedure to make your transition as smooth as possible is as follows:

1. Back up all your xml / xsl files first.
2. Read all release notes [online](#) from your current version to the latest version in correct order (from oldest release notes to newest).
3. Adapt you xsm / xsl files accordingly to the release notes.

xmlmill

5. Download

On the [download](#) page following distributions (.zip or .tar.gz format) are available:

5.1. Distributions

XMLMill for Java This distribution allows you to run XMLMill as an application, in batch or in a servlet container (it does not contain the Xalan .jars files).

XMLMill for Domino This is XMLMill for Java but a special connection program is added so XMLMill can be used on a Notes Domino R6 server.

If you download this distribution and do not already have a version of Xalan/Xerces please download the latest Xalan distribution (containing also Xerces) at: [Xalan-Java version 2](#).

Extract following files from the Xalan distribution (put these in the directory where the XMLMill distribution will be extracted).

- ◆ xml-apis.jar
- ◆ xslibc.jar
- ◆ xalan.jar
- ◆ xercesImpl.jar

Remark:

Do not forget to adapt your classpath (or the **.cmd** or **.sh** file) to include these .jar files (see below).

5.2. Extra downloads

Following extra package is available:

Signed .jar package This package contains signed versions of the needed .jars files, so you will be able to use XMLMill in an applet.

5.3. Distribution's structure

The distributions (.zip or .tar.gz format) contain following structure:

- volume** Several large .xml files containing 1000 up to 50000 records, used during stress testing (results will be published shortly on the website). The .xsl files generate one document containing the records found in the .xml files as paragraphs, textboxes or one large table. If needed, modify this xml/xsl file to incorporate your own stress tests if you want to know how performant XMLMill is.
- testservlet** Related files (and documentation) needed to get XMLMill up and running in a servlet container.
- examples** Examples directory containing different examples.

In the 'root' directory following files are found:

- xmlmill.jar** An unsigned version of the xmlmill.jar file (cannot be used in applets).
- readme.txt** The readme document that comes with the download
- gui.cmd** A batch file to start XMLMill as an application (Windows environment).
- gui.sh** A batch file to start XMLMill as an application (Linux/Unix environment).
- batch.cmd** A batch file to start XMLMill from the command-line (batch)(Windows environment).
- batch.sh** A batch file to start XMLMill from the command-line (batch)(Linux/Unix environment).

5.4. Signed jar's package structure

Following files are included:

- xmlmill.html** A batch file to start XMLMill as an applet.
- sxmlmill.jar** A signed version of the xmlmill.jar file (when XMLMill is used inside an applet).
- sxml-apis.jar** A signed version of the sxml-apis.jar file.
- sxsrtc.jar** A signed version of the sxsrtc.jar file.
- sxalan.jar** A signed version of the xalan.jar file.
- sxercesImpl.jar** A signed version of the xercesImpl.jar file.

xmlmill

6. XMLMill Interactive

This chapter describes changes regarding using XMLMill interactive.

XMLMill can be used interactively as follows:

- ◆ In a browser through the use of an applet.
- ◆ As a Java application.

Most important changes are:

1. Detached GUI window (in browser).
2. XMLMill can now be used as a stand-alone Java application.
3. Renewed interface (for improved functionality and ease of use).

6.1. XMLMill in a browser

When used with the applet the gui window is detached from the browser. The browser window can be minimized as needed.

To use XMLMill in an applet the **signed** versions of following .jar files are needed:

- ◆ **sxmlmill.jar**
- ◆ **sxml-apis.jar**
- ◆ **sxsltc.jar**
- ◆ **sxalan.jar**
- ◆ **sxercesImpl.jar**

These signed versions can be found in the **full** download package of XMLMill (see the [download](#) chapter).

You can use the **xmllmill.html** file (in a browser) to start XMLMill as an applet.

6.2. XMLMill as application

XMLMill can run as a Java application using the **com.xmlmill.applet.Main** class. Two batch files (**xmllmill.sh**, **xmllmill.sh**) are included for your convenience. Please modify these batch files to include the **.jar** files correctly in the **classpath** and modify the directory pointing to the **java.exe**.

The default content is (.cmd):

```
@echo off
REM -----
REM -- XMLMILL GUI --
REM -----

REM PLEASE modify CLASSPATH TO INCLUDE JAR FILES CORRECTLY.
```

```
REM PLEASE modify JAVA_HOME TO POINT TO JAVA.EXE.

set JAVA_HOME="C:\Program Files\JavaSoft\JRE\1.3.1_05\bin\java"
set CLASSPATH=c:\xmlmill132r4\sxercesimpl.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xalan.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xml-apis.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xmlmill.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xsltc.jar

%JAVA_HOME% com.xmlmill.applet.Main
```

and for Linux/UNIX (.sh)

```
#!/bin/bash

# -----
# -- XMLMILL GUI --
# -----

# PLEASE ADAPT CLASSPATH TO INCLUDE JAR FILEs CORRECTLY.
# PLEASE ADAPT DIR POINTING TO JAVA.EXE.

export CLASSPATH=$HOME/version135d1/xmlmillforjavatrialunix/xercesImpl.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xalan.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xml-apis.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xmlmill.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xsltc.jar

java com.xmlmill.applet.Main
```

If everything is configured correctly, following message is displayed at start-up:

```
XMLMill Java 1.35 Use the -? parameter to see options
```

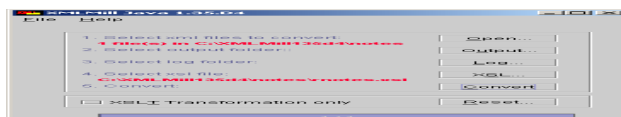
Next a splash screen will be visible while XMLMill is reading the contents of your hard-drive:



The following section will explain in detail the user interface.

6.3. Renewed interface

The GUI interface has changed for improved functionality and ease of use:



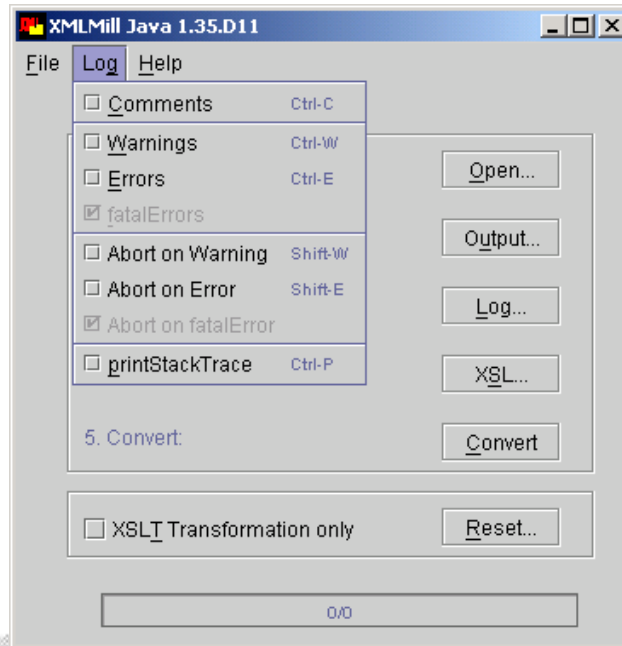
6.3.1. Menu options

6.3.1.1. File|Exit

Quit the application.

6.3.1.2. Log

The Log menu defines the different options regarding the content of the **.log** file:



Following type of information can be logged:

6.3.1.2.1. Comments

Comments consists of the text between the `<xsl:comment>` and `</xsl:comment>` text. This option can facilitate the debugging of the transformation.

6.3.1.2.2. Warnings

Log all warnings generated by the xslt transformer or XMLMill.

6.3.1.2.3. Errors

Log all errors generated by the xslt transformer or XMLMill.

6.3.1.2.4. Abort on Warning

Stop the transformation proces if a warning has occurred.

6.3.1.2.5. Abort on Error

Stop the transformation proces if an error has occurred.

6.3.1.2.6. PrintStackTrace

Print the Java Stacktrace with each **warning**, **error** or **fatalError** logged.

Remark:

The **fatalErrors** and **Abort on fatalError** are always activated. A fatal error will always be logged and will always abort the generation of the .pdf file.

6.3.1.3. Help|About

Shows following **About** screen:



6.4. Buttons

6.4.1. Open...

The filechooser window is opened and the appropriate .xml files can be selected.

6.4.2. Output...

The filechooser window is opened and the appropriate output directory can be selected. This selection will overrule the outputfolder defined in the .xml or .xsl file (if any)

6.4.3. Log...

The filechooser window is opened and the appropriate .xml files can be selected.

6.4.4. XSL...

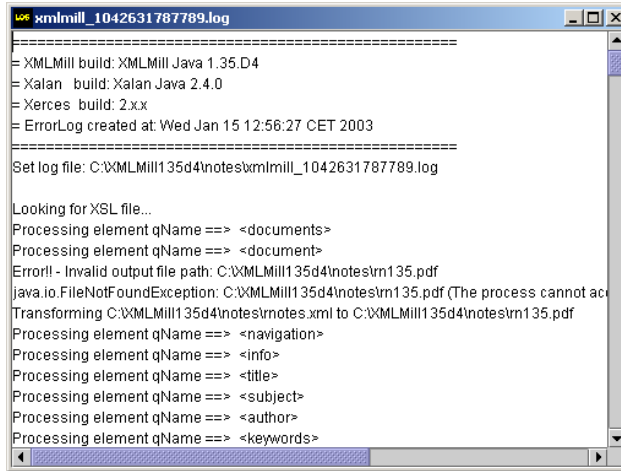
The filechooser window is opened and the appropriate .xsl file can be selected. Only one .xsl file can be selected. This .xsl file is used to transform the .xml files. This .xsl file will overrule the .xsl defined in the .xml file (if any).

6.4.5. Convert...

Converts the selected xml files to PDF. The .pdf files are written in the output directory (if one selected). The name of the .pdf file is defined by the **file** attribute in the **<document>** tag mentioned in the .xml or

.xsl file.

After conversion the .log file is displayed in a separate window:



The log file will always display following information:

```

=====
= XMLMill build: XMLMill Java 1.35.D4
= Xalan   build: Xalan Java 2.4.0
= Xerces  build: 2.x.x
= ErrorLog created at: Wed Jan 15 12:56:27 CET 2003
=====
    
```

- ◆ The XMLMill build information.
- ◆ The Xalan build version used.
- ◆ The Xerces build version used.
- ◆ The date and time the errorlog was created.

Depending on the .log options chosen other information is also logged.

6.4.6. Reset...

Resets all fields.

6.4.7. XSLT Transformation only

When this option is selected a transformation file is generated for each xml transformation. This file is stored in the output directory where normally the pdf file is stored. It will have the same name as the xml file but prefixed with **t_**. This file can be used to check the transformation output against the xmlmill.dtd (using a xml editor that can validate a .xml file against a dtd).

This option is interesting when you generate a pdf document with a xml and xsl file. With this option you can inspect what the transformation result is.

7. XMLMill batch

XMLMill can be used in batch, this means 'off-line' using command-line parameters. This way XMLMill can be used in a scheduled environment where a scheduler calls XMLMill passing the correct parameters.

Most important changes are:

1. Support for 'headless' environments.
2. New parameters to support new functionality.

7.1. Support for 'headless' environments

As of this version XMLMill can be used in a 'headless' environment. This means an environment without a X server installed (which is true for many Linux/UNIX server systems).

XMLMill can be started in batch using the **com.xmlmill.batch.Main** class. Two batch files (**xmlmill.sh**, **xmlmill.cmd**) are included for your convenience. Please modify these batch files to include the **.jar** files correctly in the **classpath** and modify the directory pointing to the **java.exe**.

The default content is (.cmd):

```
@echo off
REM -----
REM -- XMLMILL BATCH --
REM -----

REM PLEASE modify CLASSPATH TO INCLUDE JAR FILES CORRECTLY.
REM PLEASE modify DIR POINTING TO JAVA.EXE.

set JAVA_HOME="C:\Program Files\JavaSoft\JRE\1.3.1_05\bin\java"
set CLASSPATH=c:\xmlmill132r4\xercesimpl.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xalan.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xml-apis.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xmlmill.jar
set CLASSPATH=%CLASSPATH%;c:\xmlmill132r4\xslt.jar

@echo on

%JAVA_HOME% com.xmlmill.batch.Main -?
```

and for Linux/UNIX (.sh)

```
#!/bin/bash

# -----
# -- XMLMILL BATCH --
# -----

# PLEASE ADAPT CLASSPATH TO INCLUDE JAR FILES CORRECTLY.
# PLEASE ADAPT DIR POINTING TO JAVA.EXE.

export CLASSPATH=$HOME/version135d1/xmlmillforjavatrialunix/xercesImpl.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xalan.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xml-apis.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xmlmill.jar
export CLASSPATH=$CLASSPATH:$HOME/version135d1/xmlmillforjavatrialunix/xslt.jar

java com.xmlmill.applet.Main
```

The script calls Java and the **com.xmlmill.batch.Main** class with the appropriate flags and arguments (described below). The following command line, for example, generates for all .xml files found in

c:\xmlmill135\xmlxsl a .pdf document using the **invoices.xsl** file:

```
%JAVA_HOME% com.xmlmill.applet.XMLMill -m c:\xmlmill135\xmlxsl -l  
c:\xmlmill135\xmlxsl\invoices.xsl
```

Please find below an overview of the flags and arguments:

- ? Prints the on-line help message.
- v Enable verbose output.
- m Defines the xmlfile or directory containing xml files. Multiple -m files can be defined.
- s Defines the xslfile that will be used to transform the xml files. If specified this .xsl file will overrule any .xsl file defined in the .xml file (defined in the <?xml-stylesheet ...?> tag).
- l Defines the (existing) logfolder and/or logfile name. If a file-name is defined it should end with **.log**. If the log-folder only contains a filename (ending in **.log** the log will be written in directory of first xml-file that is processed. If no log-folder is defined, no log-folder will be written. All log messages (if any) will be send to the screen.
- o Defines the (existing) outputfolder. If no outputfolder is defined, the output will be written in the directory defined by **file** attribute of the <output> tag of the first xml-file processed.

7.2. New parameters to support new functionality

Following parameters were added to reflect the new functionality:

- a Append messages to existing log file instead of overwriting the log file.
- c Add comments to the log file.
- e Defines which errors (warnings, errors or fatalErrors) should be logged and on which errors XMLMill should abort generation of the pdf document. This parameter is the sum (combination) of following values: (1) log errors (2) abort on errors (4) log warnings (8) abort on warnings (16) print stacktrace. Note that a **fatalError** will always be logged and will lead to aborting the generation.
- t XSLT transformation only.

8. XMLMill in Servlets

XMLMill can be used in servlets, using the **PDXAPI**. The **PDXAPI** contains only two classes. For more information on how to achieve this, please visit [our website](#) and our online [Javadoc API](#).

In this chapter we will only describe the changes of **PDXAPI** in comparison with the previous version.

8.1. Class: PDXTransform

Normally, you will configure your **PDXTransform** instance to use an instance of the **PDXTransformErrorHandler** class to process faults that are uncovered during parsing (or during generation of the .pdf document). This is done with the **PDXTransform.setErrorHandler()** method.

If a separate errorhandler instance is not added to the **PDXTransform** instance, the default errorhandler will be used. This default errorhandler has following behavior:

- ◆ **FatalErrors** are **always** logged and aborted.

8.2. Class: PDXTransformErrorHandler

This class implements the **org.xml.sax.ErrorHandler** interface.

This way all errors that occur during .pdf generation follow the same logic as the **org.xml.sax.ErrorHandler**.

With this class you only need one errorhandler instance to detect parsing errors (thrown by the SAX handler beneath XMLMill) and .pdf generation errors (thrown by XMLMill). Following methods are available:

8.2.1. **public PDXTransformErrorHandler()**

This is the default constructor that mimics the default behavior of a SAX2 compliant parser. This means:

- ◆ **FatalErrors** are **always** logged and aborted.

8.2.2. **public PDXTransformErrorHandler(int flags)**

Constructs an instance of this class and define the behavior using the constants defined in this class:

PRINT_ERROR	Log recoverable parser errors (value = 1).
ABORT_ON_ERROR	Abort when a recoverable parser error has occurred (value = 2).
PRINT_WARNING	Log warnings (value = 4).
ABORT_ON_WARNING	Abort when a warning has occurred (value = 8).
PRINT_STACKTRACE	Print Java's printStackTrace when a warning, recoverable error or fatal Error has occurred (value = 16).

The flags parameter is the sum of above values, indicating which errors to log and to abort on.

8.2.3. void error (SAXParseException e)

This method is used to report errors that aren't expected to be fatal. The best-known example is violation of XML validity constraints. Not many application-level errors (as reported by XMLMill) will fall into this category as XMLMill's errors will be **warnings** or **fatalErrors**.

8.2.4. void fatalError (SAXParseException e)

This method is used to report errors, typically violations of well formedness, that are fatal. Most application-level errors (as reported by XMLMill) will fall into this category. Parsing (and therefore generation of the .pdf document) is aborted.

8.2.5. void warning (SAXParseException e)

This method is used to report warnings. Unlike fatal and nonfatal errors, the XML specification doesn't place requirements on reporting such situations. XMLMill will generate warnings when attributes have the wrong value or attributes are not recognized.

xmlmill

9. XMLMill in applications

The use of the **PDEAPI** has been discontinued. With the **PDEAPI** a PDF document can be generated using **PDEElements** objects. A **PDEElements** object is a representation of an 'object' on a page, as header, footer, textbox, paragraph, image, watermark, font,

The next version of XMLMill a new API will be presented facilitating the way XMLMill can be used in Java applications (through the use of propertyLists).

Users who use the current **PDEAPI** should continue to use the previous release of XMLMill until the next version is available.

xmlmill

10. Images

10.1. Image processing

The image class model was refactored to optimize the way images are read from a file, stored in memory and written to the PDF document:

Read from a file:

The speed images are written to the PDF document has been optimized. Performance increase from 10 to 50 % should be possible.

Memory storage:

No more than one image will be held in memory at a time. An image will only be in memory the time the image is written to the PDF document. The memory occupied will be released once the image is written in the stream. This means that XMLMill is now suited to generate documents with a huge numbers of images.

Writing to the PDF document:

The bytes are immediately compressed so huge image files will consume considerably less memory (up to 90% compression - depending on image type).

10.2. New image types:

This version of XMLMill has been extended to support more types of images. The supported image types are:

- ◆ PCX - PC Paintbrush
- ◆ BMP - Windows Bitmap
- ◆ DWG - AutoCAD R13/R14/R2000 DWG File
- ◆ GIF - Compuserve GIF
- ◆ JPG - JPEG Image

10.2.1. PCX - PC Paintbrush

The .pcx file format was developed by the ZSoft Corporation for the PC Paintbrush program in the early 80's. It is a very simple format that uses Run Length Encoding (RLE) to compress the image data.

XMLMill supports following .pcx formats:

- ◆ 2 colors (1-bit)
- ◆ 16 color (4-bit)
- ◆ 256 colors (8-bit)

- ◆ True color (24-bit)

Example

```
<image src="images\lion.pcx" />
```

10.2.2. BMP - Windows Bitmap

This file format is the MS-Windows standard format. It holds black&white-, 16-color, 256-color and Truecolor images. The palletized 16-color and 256-color images may be compressed via run length encoding. Notice there is also a OS/2-BMP format

XMLMill supports following .pcx formats:

- ◆ 2 colors (1-bit)
- ◆ 16 color (4-bit)
- ◆ 16 color (4-bit-rl)
- ◆ 256 colors (8-bit)
- ◆ 256 colors (8-bit-rl)
- ◆ 256 colors (16-bit)
- ◆ True color (24-bit)
- ◆ True color (32-bit)

Example

```
<image src="images\lion.bmp" />
```

10.2.3. DWG - AutoCAD R13/R14/R2000 DWG File

XMLMill is capable of retrieving out of the DWG file the thumbnail image.

XMLMill supports following .dwg formats:

- ◆ Autocad R13 dwg files.
- ◆ Autocad R14 dwg files.
- ◆ Autocad R2000 dwg files.

Example

```
<image src="images\lion.dwg" />
```

11. New tags

This section chapter gives an overview of all new tags in this version. For an overview of the changes of the existing tags please read chapter [Existing tags](#).

11.1. <navigation>

Description:

The <navigation> tag controls whether the document should contain a table of contents and/or outlines and defines the numbering format of the levels displayed in the table of contents and/or outlines.

Appears in:

<document> tag

Example:

```
<navigation number-format="1." outlines="on" table-of-contents="on" />
```

11.1.1. Attribute: **number-format**

The **number-format** attribute defines how the number should be formatted.

Values

- A** Format number according to the alphabet (uppercase): A,B,C,D,...
- a** Format number according to the alphabet (lowercase): a,b,c,d,...
- I** Format number using Roman numbers (uppercase): I,II,III,IV,...
- i** Format number using Roman numbers (lowercase): i,ii,iii,iv,...
- 0 or 1** Format number using Decimal numbers with leading zero's.
- I.A.1** Combine previous formats to a 'combined number'. Characters other than A,a,I,i,0,1 are interpreted as dividers.

11.1.2. Attribute: **outlines**

The **outlines** attribute defines if the document should be generated with outlines (bookmarks).

Values

- on** Generate the document with outlines (bookmarks).
- off** Do not generate outlines (bookmarks) (default)

11.1.3. Attribute: **table-of-contents**

The **table-of-contents** attribute defines if the document should be generated with table-of-contents.

The table-of-contents are generated on the page where this tag is set. A table-of-contents will always start on a new page, inserting subsequent pages if necessary to print the table-of-contents. Please visit the

[table-of-contents](#) section for more information.

Values

on Generate the document with a table-of-contents.

off Do not generate a table-of-contents.

11.2. <viewerpreferences>

Description:

The **<viewerpreferences>** tag controls the way the document is to be displayed on the screen. If this entry is absent, viewer applications should use their own current user preference settings.

Appears in:

<document> tag

Example:

```
<document page-mode="useoutlines"
  centerwindow = "on"
  fitwindow    = "on"
  hidemenubar  = "off"
  hidetoolbar  = "off"
  hidewindowui = "on">
```

11.2.1. Attribute: centerwindow

The **centerwindow** attribute defines whether to position the document's window in the center of the screen.

Values

on Position the document's window in the center of the screen.

off Do not position the document's window in the center of the screen (default).

11.2.2. Attribute: fitwindow

The **fitwindow** defines whether to resize the document's window to fit the size of the first displayed page.

Values

on Resize the document's window to fit the size of the first displayed page.

off Do not resize the document's window to fit the size of the first displayed page(default).

11.2.3. Attribute: hidemenubar

The **hidemenubar** defines whether or not to hide the viewer application's menu bar when the document is active.

Values

on Hide the viewer application's menu bar when the document is active.

off Do not hide the viewer application's menu bar when the document is active (default).

11.2.4. Attribute: **hidetoolbar**

The **hidetoolbar** defines whether to hide the viewer application's toolbars when the document is active.

Values

on Hide the viewer application's toolbar when the document is active.

off Do not hide the viewer application's toolbar when the document is active (default).

11.2.5. Attribute: **hidewindowui**

The **hidewindowui** defines whether to hide user interface elements in the document's window (such as scroll bars and navigation controls), leaving only the document's contents displayed.

Values

on Hide user interface elements in the document's window.

off Do not hide user interface elements in the document's window (default).

11.3. **<table-of-contents>**

Description:

The **table-of-contents** tag defines the characteristics of the table of contents. On the place this tag is put the toc will be generated.

Appears in:

<content> tag

Example:

```
<table-of-contents show-levels="2" show-page-number="on" leader-pattern=". "
right-align-page-number="on" >
<toc-header link-id="IDAKAIU" font-size="24pt" align="center" >Table of Contents
<toc-header>
<toc-style level="1" font-size="14pt" font-style="bold" padding-top="0.5cm" />
<toc-style level="2" font-size="12pt" font-style="normal" padding-left="0.3cm" />
</table-of-contents>
```

11.3.1. Attribute: **show-levels**

Use this attribute to define how many levels should be displayed in the table-of-contents. If this attribute is not set, all levels found in the document are mentioned in the table-of-contents.

Values

1-9 A table-of-contents can contain up to 9 levels.

11.3.2. Attribute: **show-pagnumber**

Use this attribute to define if the page-number should be shown in the table-of-contents.

Values

on Display the page-numbers.

off Do not display page-numbers..

11.3.3. Attribute: **right-align-page-number**

Use this attribute to define if the page-number should be right aligned in the table-of-contents.

Values

on Right align the page-numbers.

off Do not right align page-numbers (the page-numbers are put immediately after the text).

11.3.4. Attribute: **leader-pattern**

Use this attribute to define the pattern of characters that should be printed between the text and the page-number.

This pattern can consist of more characters and any character is valid. Typically, a leader-pattern like '.' or '!' will be used.

11.3.5. **<toc-header>**

Use the **<toc-header>** tag to define a heading for the table-of-contents. Please check the xmlmill.dtd to know which attributes can be used with this tag.

11.3.6. **<toc-style>**

With the **<toc-style>** tag the formatting of the different levels in the table-of-contents are defined. Please check the xmlmill.dtd to know which attributes can be used with this tag.

11.3.7. Attribute: **level**

Use this attribute to define the level this style is applicable on. This is a required attribute.

Values

1-9 A table-of-contents can contain up to 9 levels.

11.4. <list-numbers>

Description:

The **<list-numbers>** tag defines an ordered list.

Appears in:

<content> tag

Example:

```
<list-numbers number-format="1." label-indent="0cm" text-indent="1.0cm" start-at="3">  
<list-label-format font-style="bold" />  
<list-item-text>Item 1.</list-item-text>  
<list-item-text>Item2.</list-item-text>  
<list-item-text>Item3.</list-item-text>  
</list-numbers>
```

result:
3. Item 1.
4. Item 2.
5. Item 3.

11.4.1. Attribute: **number-format**

Defines the format of the numbering to use. For more information on this attribute, please visit the [number-format](#) section in the navigation section.

11.4.2. Attribute: **label-indent**

Defines the indentation of the label (this means the indentation of the number).

11.4.3. Attribute: **text-indent**

Defines the indentation of the list-item-text. If the value **auto** is defined, indentation is based on the width of all the **list-label** tags in this list.

11.4.4. Attribute: **start-at**

Defines a new starting number for this list. If not defined, the list starts at number 1.

11.4.5. **<list-label-format>**

Defines the format of the numbers. Please check the xmlmill.dtd to know which attributes can be used with this tag.

11.4.6. **<list-item-text>**

The tag representing a text-item in the list. Please check the xmlmill.dtd to know which attributes can be used with this tag.

11.5. **<list-bullets>**

Description:

The **<list-bullets>** tag defines an unordered list.

Appears in:

<content> tag

Example:

```
<list-bullets label-indent="0cm" text-indent="auto">  
<list-label>XML</list-label>  
<list-item-text>XML document</list-item-text>  
<list-label>XSL</list-label>  
<list-item-text>XSL Stylesheet</list-item-text>
```

```
<list-label>XSLT</list-label>
<list-item-text>Extensible Stylesheet Language</list-item-text>
</list-bullets>
result:
XML XML document.
XSL XSL Stylesheet
XSLT Extensible Stylesheet Language
```

11.5.1. Attribute: **label-indent**

Defines the indentation of the label (this means the indentation of the label).

11.5.2. Attribute: **text-indent**

Defines the indentation of the list-item-text. If the value **auto** is defined, indentation is based on the width of all the **list-label** tags in this list.

11.5.3. **<list-label>**

The **list-label** tag defines the label to be used in this unordered list. This can be used to define the 'bullet-style' to use. Please note that the bullet-style can also be regular text and that a different bullet-style can be defined per list-item.

11.5.4. **<list-item-text>**

The tag representing a text-item in the list. Please check the xmlmill.dtd to know which attributes can be used with this tag.

11.6. **<page-number>**

Description:

The **<page-number>** tag represents a unique label (the pagenumber) to the current page. As actual pagenumbers cannot be known until XMLMill has 'paginated' the source text, the position where the pagenumber should appear must be specified using the **<page-number>** tag.

Appears in:

<header>, **<footer>**, **<textbox>**, **<inline>** tag

Example:

```
<header align="left"><inline font-style="bold"><page-number/></inline></header>
```

11.7. **<page-total>**

Description:

The **<page-total>** tag represents the total number of pages in the document. As the number of pages cannot be known until XMLMill has 'paginated' the source text, the position where the pagenumber total should appear must be specified using the **<page-total>** tag.

Appears in:

<header>, <footer>, <textbox>, <inline> tag

Example:

```
<header align="left"><inline font-style="bold"><page-total/></inline></header>
```

11.8. <file>**Description:**

The <file> tag represents the name of the generated .pdf document, including the directory where the file resides.

If the .pdf document is generated using a stream, the filename does not exist and this tag will generate an empty value.

Appears in:

<header>, <footer>, <textbox>, <inline> tag

Example:

```
<header align="left"><inline font-style="bold"><file/></inline></header>
```

11.9. <date>**Description:**

The <date> tag represents the date at the moment of generating the .pdf document. The format is based on the current Locale (defined by the machine's OS settings).

To define a specific format for a date, use the <datetime> tag.

Appears in:

<header>, <footer>, <textbox>, <inline> tag

Example:

```
<header align="left"><inline font-style="bold"><date/></inline></header>
```

11.10. <time>**Description:**

The <time> tag represents the time at the moment of generating the .pdf document. The format is based on the current Locale (defined by the machine's OS settings).

To define a specific format for a date, use the <datetime> tag.

Appears in:

<header>, <footer>, <textbox>, <inline> tag

Example:

```
<header align="left"><time/></header>
```

11.11. <datetime>

Description:

With the <datetime> tag the format of the required date and/or time can be defined:

Appears in:

<header>, <footer>, <textbox>, <inline> tag

Example:

```
<datetime format="yyyy.MM.dd G 'at' hh:mm:ss z"/>
Result:
2003.02.05 AD at 15:08:56 PDT
```

11.11.1. Attribute: format

With the **format** attribute the time format can be defined using a time pattern string. In this pattern, all ASCII letters are reserved as pattern letters, which are defined as the following:

Symb.	Meaning	Presentation	Example
G	era designator	(Text)	AD
y	year	(Number)	1996
M	month in year	(Text & Number)	July & 07
d	day in month	(Number)	10
h	hour in am/pm (1~12)	(Number)	12
H	hour in day (0~23)	(Number)	0
m	minute in hour	(Number)	30
s	second in minute	(Number)	55
S	millisecond	(Number)	978
E	day in week	(Text)	Tuesday
D	day in year	(Number)	189
F	day of week in month	(Number)	2 (2nd Wed in July)
w	week in year	(Number)	27
W	week in month	(Number)	2
a	am/pm marker	(Text)	PM
k	hour in day (1~24)	(Number)	24
K	hour in am/pm (0~11)	(Number)	0
z	time zone	(Text)	Pacific Standard Time
'	escape for text	(Delimiter)	
''	single quote	(Literal)	'

The count of pattern letters determines the format:

(Text): 4 or more pattern letters--use full form, < 4--use short or abbreviated form if one exists.

(Number): the minimum number of digits. Shorter numbers are zero-padded to this amount. Year is handled specially; that is, if the count of 'y' is 2, the Year will be truncated to 2 digits.

(Text & Number): 3 or over, use text, otherwise use number.

Any characters in the pattern that are not in the ranges of ['a'..'z'] and ['A'..'Z'] will be treated as quoted text. For instance, characters like ':', ',', '#', and '@' will appear in the resulting time text even they are not embraced within single quotes. A pattern containing any invalid pattern letter will result in a thrown exception during formatting or parsing.

Examples Using the US Locale:

Format Pattern	Result
"yyyy.MM.dd G 'at' hh:mm:ss z"	1996.07.10 AD at 15:08:56 PDT
"EEE, MMM d, 'yy"	Wed, July 10, '96
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:00 PM, PST
"yyyyy.MMMMM.dd GGG hh:mm aaa"	1996.July.10 AD 12:08 PM

11.12. <link>

Description:

Electronic documents are able to contain 'hot text' that acts as a link to other items in the same document or to items in another document. Web browsers commonly highlight such 'hypertext' links using underlined, coloured styles. The **<link>** tag is used to create such a link:

Appears in:

<p>, **<textbox>**, **<list-item-text>**, **<inline>**, **<field>** tag

Example:

```
<link external-destination="http://www.xmlmill.com">
```

```
<link internal-destination="chapter1">
```

11.12.1. Attribute: **internal-destination**

An internal destination is identified by the **link-id** attribute on the other object.

11.12.2. Attribute: **external-destination**

An external destination is identified by a URL.

12. Existing tags

This chapter describes the changes of the behavior or attributes of existing tags (this means: tags present in a previous version).

Remark:

Please note that this chapter only describes the new attributes added to existings tags. For an overview of all attributes of a tag, please visit our [website](#)."

12.1. <documents>

The **<documents>** tag has been extended with attributes to control the way white-spaces, spaces and linefeeds are treated when reading xml data.

This way the 'format' of the xml document (this means: the existance of whitespace inside tags) has no influence on the formatting of the PDF document regarding linefeeds, duplicate spaces and tabs.

In order to preserve compatibility with previous versions of XMLMill (regarding the appearance of spaces, linefeeds and tabs in xml documents), following are the default settings of these attributes:

whitespace-collapse off
space-collapse off
space-treatment preserve
linefeed-treatment preserve

12.1.1. Attribute: **whitespace-collapse**

When there are consecutive whitespace characters this may be due to the formatting of the source xml document rather than any intended document content.

The **whitespace-collapse** attribute allows to reduce the sequences of whitespace characters down to a single space character.

A whitespace is determined according to the Java rules. A character is considered to be a Java whitespace character if and only if it satisfies one of the following criteria:

- ◆ It is a Unicode space separator (category "Zs" in the Unicode specification data file), but is not a no-break space (\u00A0 or \uFEFF).
- ◆ It is a Unicode line separator (category "Zl").
- ◆ It is a Unicode paragraph separator (category "Zp").
- ◆ It is \u0009, HORIZONTAL TABULATION
- ◆ It is \u000A, LINE FEED.
- ◆ It is \u000B, VERTICAL TABULATION
- ◆ It is \u000C, FORM FEED

- ◆ It is \u000D, CARRIAGE RETURN
- ◆ It is \u001C, FILE SEPARATOR
- ◆ It is \u001D, GROUP SEPARATOR.
- ◆ It is \u001E, RECORD SEPARATOR.
- ◆ It is \u001F, UNIT SEPARATOR

Values

on Turns white-space collapse on.

off Turns white-space collapse off.

Example

```
Code:
<documents whitespace-collapse="on">
...
<p>This      text      is      widely      spaced. </p>
...
</documents>

Result:
This text is widely spaced.
```

```
Code:
<documents whitespace-collapse="of">
...
<p>This      text      is      widely      spaced. </p>
...
</documents>

Result:
This      text      is      widely      spaced.
```

12.1.2. Attribute: **space-collapse**

The **space-collapse** attribute reduces the sequences of spaces (or tabs) down to a single space character. A character is considered to be a space character if and only if it is specified to be a space character by the Unicode 2.0 standard (category "Zs", "Zl, or "Zp" in the Unicode specification data file), or is a horizontal tabulator character (\u0009).

The difference with the **whitespace-collapse** attribute is that the **space-collapse** attribute only reduces the sequences of spaces or tabs, not linefeed characters.

You can use this attribute if you want to preserve the line-feed settings in a tag.

Values

on Turns space collapse on.

off Turns space collapse off.

Example

```
Code:
<documents space-collapse="on">
...
<p>This      text  is  widely
spaced. </p>
...
</documents>

Result:
This text is widely
spaced.
```

12.1.3. Attribute: **space-treatment**

The **space-treatment** attribute can be used to remove all whitespace characters, except for linefeed characters (which are described below). This attribute therefore affect the space character and the tab character.

Values

preserve Keeps al spaces and tabs as defined in the data (no removal of duplicate spaces) (default).

ignore Removas duplicate spaces and tabs.

Example

```
Code:
<documents space-treatment="on">
...
<p>This      text  is  widely
spaced. </p>
...
</documents>

Result:
Thistextiswidely
spaced.
```

12.1.4. Attribute: **linefeed-treatment**

The **linefeed-treatment** attribute is used to 'preserve' or 'ignore' linefeed characters, but adds the 'treat-as-space' option.

Values

preserve Instructs XMLMill to treat line feeds as line-ending instructions (to place the next character on the following line).

ignore Ignore the linefeed characters.

treat-as-space Treat line feeds as a normal space character.

Example

```
Code:
<documents linefeed-treatment="ignore">
<p>Line one.
Line two.
Line three. </p>
</documents>

Result:
```

```
Line one.Line two.Line three.
```

```
Code:
<documents linefeed-treatment="preserve">
<p>Line one.
Line two.
Line three. </p>
</documents>
```

```
Result:
Line one.
Line two.
Line three.
```

```
Code:
<documents linefeed-treatment="treat-as-space">
<p>Line one.
Line two.
Line three. </p>
</documents>
```

```
Result:
Line one. Line two. Line three.
```

12.2. <document>

The **<document>** has been extended with attributes to control how the document should be displayed on the screen, such as whether its outline and thumbnail page images should be displayed automatically and whether some location other than the first page should be shown when the document is opened.

12.2.1. Attribute: **page-mode**

The **page-mode** attribute defines how the document should be displayed when opened.

Values

usenone Neither document outline nor thumbnail images visible.

useoutlines Document outline visible.

fullscreen Full-screen mode, with no menu bar, window controls, or any other window visible.

sethumbs Thumbnail images visible.

Example

```
<document page-mode="useoutlineslapse">
```

12.2.2. Attribute: **page-layout**

The **page-layout** tag defines which the page layout to be used when the document is opened:

Values

singlepage Display one page at a time.

onecolumn Display the pages in one column

twocolumnleft Display the pages in two columns, with oddnumbered pages on the left.

twocolumnright Display the pages in two columns, with oddnumbered pages on the right.

Example

```
<document page-layout="onecolumn">
```

12.3. <info>

The **date** attribute has been deprecated. Please remove this attribute from your .xml/.xsl files if this attribute has been used.

12.4. <p>

12.4.1. Attribute: **hanging-indent**

The **hanging-indent** attribute will put the second and following lines in a paragraph at the defined position.

```
<p hanging-indent="2cm">The hanging-indent attribute will put the second and following  
lines in a paragraph at the defined position.</p>
```

12.4.2. Attribute: **auto-numbering**

The **auto-numbering** attribute is used to mark that the paragraph's heading number will automatically be generated by XMLMill. This attribute is only effective if the [<navigation>](#) tag is used in the **<page-template>** attribute.

The number will have the format as defined by the number-format attribute defined by the **<navigation>** tag.

```
<p auto-numbering="on">Some heading</p>
```

12.4.3. Attribute: **outline-level**

The **auto-numbering** attribute defines at which level the number should be generated. For example: The number: **A.II.9** has three levels. If outline-level is two, the number will be increased at the second level. This means the the paragraph will get number **A.III.1**

```
<p outline-level="2">Some heading</p>
```

This attribute is only effective if the **<navigation>** tag is used in the **<page-template>** attribute.

12.4.4. Attribute: **show-in-outlines**

The **show-in-outlines** attribute defines if the paragraph needs to be mentioned in the outlines of the .pdf document. The whole content of the paragraph will be copied into the document's outlines.

This attribute is only effective if the **<navigation>** tag is used in the **<page-template>** attribute.

```
<p show-in-outlines="2">Some heading</p>
```

12.4.5. Attribute: **show-in-toc**

The **show-in-toc** attribute defines if the paragraph needs to be mentioned in the document's table-of-contents. The whole content of the paragraph will be copied into the document's outlines.

This attribute is only effective if the **<table-of-contents>** tag is used.

```
<p show-in-outlines="2">Some heading</p>
```

12.4.6. Attribute: **link-id**

The **link-id** attribute defines a link-id value to the paragraph. This link-id value can be used as a reference in a **internal-destination** attribute in the [link](#) tag.

Care should be taken that each link-id is uniquely defined.

```
<p link-id="p1">Some text</p>

Somewhere in the document the link-id can be used in a <link> tag:
<link internal-destination="p1"></link>
```

12.4.7. Attribute: **linefeed-treatment**

See the [linefeed-treatment](#) section for more information.

12.4.8. Attribute: **space-collapse**

See the [space-collapse](#) section for more information.

12.4.9. Attribute: **space-treatment**

See the [space-treatment](#) section for more information.

12.4.10. Attribute: **whitespace-collapse**

See the [whitespace-collapse](#) section for more information.

12.4.11. Attribute: **script-mode**

The **script-mode** attribute defines if the text is to be printed in superscript or subscript.

```
<p script-mode="super">Print this all in superscript</p>
```

12.5. **<textbox>**

Alle the attributes mentioned for the [<p>](#) tag are also valid the **<textbox>** tag.

12.6. **<image>**

12.6.1. Attribute: **alternate**

This attribute is indented for later usage. Please do not use it at this time.

12.7. <inline>

12.7.1. Attribute: [linefeed-treatment](#)

See the [linefeed-treatment](#) section for more information.

12.7.2. Attribute: [space-collapse](#)

See the [space-collapse](#) section for more information.

12.7.3. Attribute: [space-treatment](#)

See the [space-treatment](#) section for more information.

12.7.4. Attribute: [whitespace-collapse](#)

See the [whitespace-collapse](#) section for more information.

12.7.5. Attribute: [script-mode](#)

See the [script-mode](#) section for more information.

12.8. <header>

12.8.1. Attribute: [border-bottom](#)

A shorthand property for setting the width, color, and style for for the bottom border in one declaration.

This attribute has exactly three values:

For more information, please see the sections below.

```
<header align="left" border-bottom="1px solid red">left header</>
```

12.8.2. Attribute: [border-bottom-color](#)

This property sets the color of the bottom border.

Values

- ◆ A hexadecimal number representing the color.
- ◆ A Java named color.
- ◆ A Java rgb color (ref, green, blue).
- ◆ The word **transparent**, to make the underlying colors shine through.

```
<header align="left" border-bottom-color="blue">left header</>
```

12.8.3. Attribute: [border-bottom-style](#)

A shorthand property for setting the style of the bottom border.

Values

solid A solid border.

none No border.

```
<header align="left" border-bottom-style="solid">left header</>
```

12.8.4. Attribute: **border-bottom-width**

This attribute sets the width of the bottom border.

Values

thin A thin border.

medium A medium border.

thick A thick border.

99/99px/99cm/99in/99mm A value indicating the width of this element expressed in pixels, centimeters, inches or millimeters (if no units of measurement is define it is defaulted to pixels).

Remark:

```
<header align="left" border-bottom-width="thin">left header</>
```

12.8.5. Attribute: **linefeed-treatment**

See the [linefeed-treatment](#) section for more information.

12.8.6. Attribute: **space-collapse**

See the [space-collapse](#) section for more information.

12.8.7. Attribute: **space-treatment**

See the [space-treatment](#) section for more information.

12.8.8. Attribute: **whitespace-collapse**

See the [whitespace-collapse](#) section for more information.

12.9. <footer>

12.9.1. Attribute: **border-top**

A shorthand property for setting the width, color, and style for for the bottom top in one declaration. This attribute has exactly three values:

For more information, please see the sections below.

```
<footer align="left" border-top="1px solid red">left footer</>
```

12.9.2. Attribute: **border-top-color**

This property sets the color of the top border.

Values

- ◆ A hexadecimal number representing the color.
- ◆ A Java named color.
- ◆ A Java rgb color (ref, green, blue).
- ◆ The word **transparent**, to make the underlying colors shine through.

```
<footer align="left" border-top-color="blue">left footer </>
```

12.9.3. Attribute: **border-top-style**

A shorthand property for setting the style of the top border.

Values

solid A solid border.

none No border.

```
<footer align="left" border-top-style="solid">left footer </>
```

12.9.4. Attribute: **border-top-width**

This attribute sets the width of the topborder.

Values

thin A thin border.

medium A medium border.

thick A thick border.

99/99px/99cm/99in/99mm A value indicating the width of this element expressed in pixels, centimeters, inches or millimeters (if no units of measurement is define it is defaulted to pixels).

Remark:

```
<footer align="left" border-bottom-width="thin">left footer </>
```

12.9.5. Attribute: **linefeed-treatment**

See the [linefeed-treatment](#) section for more information.

12.9.6. Attribute: **space-collapse**

See the [space-collapse](#) section for more information.

12.9.7. Attribute: **space-treatment**

See the [space-treatment](#) section for more information.

12.9.8. Attribute: **whitespace-collapse**

See the [whitespace-collapse](#) section for more information.

12.9.9. Attribute: **script-mode**

12.10. <thead>

12.10.1. Attribute: **hanging-indent**

See the [hanging-indent](#) attribute in the <p> section for more information.

12.11. <rows>

12.11.1. Attribute: **hanging-indent**

See the [hanging-indent](#) attribute in the <p> section for more information.

12.12. <row>

12.12.1. Attribute: **hanging-indent**

See the [hanging-indent](#) attribute in the <p> section for more information.

12.13. <field>

12.13.1. Attribute: **auto-numbering**

See the [auto-numbering](#) attribute in the <p> section for more information.

12.13.2. Attribute: **link-id**

See the [link-id](#) attribute in the <p> section for more information.

12.13.3. Attribute: **outline-level**

See the [outline-level](#) attribute in the <p> section for more information.

12.13.4. Attribute: **show-in-outlines**

See the [show-in-outlines](#) attribute in the <p> section for more information.

12.13.5. Attribute: **show-in-toc**

See the [show-in-toc](#) attribute in the <p> section for more information.

13. Known Bugs

The bugs below will be solved in subsequent revisions in the next weeks.

1. A relative url like **file:/rnotes.xml** in the 'external-destination' attribute in a<link> tag is not correctly interpreted.
2. BMP images are printed up-site-down on a page with **orientation = landscape**.
3. The euro symbol is not correctly generated.
4. Number formats with leading zero's like **number-format="001."** does not correctly generate the sequence numbers.
5. 16 bit BMP images are not correctly displayed (wrong color conversion).
6. The row-spanning of table-rows did not pass the acceptance testing.
7. The table-of-contents header cannot yet be included in the outlines (bookmarks).

xmlmill