

---

XMLMill

---

Release Notes

**XMLMill v 1.52**

**1.0**

xmlmill

© 2000-2003 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.

# Table of Contents

<b>1. Preface</b> .....	<b>5</b>
<b>2. Upgrading from previous versions</b> .....	<b>6</b>
2.1. Upgrading from version 1.51 .....	6
2.2. Upgrading from earlier versions .....	6
<b>3. What's new ?</b> .....	<b>7</b>
<b>4. Java API</b> .....	<b>8</b>
4.1. Class: PDXTransform .....	8
4.1.1. Method: setStylesheetParam(String key, Object expression) .....	8
<b>5. PNG Images</b> .....	<b>10</b>
<b>6. Bugfixes</b> .....	<b>11</b>

xmlmill

xmlmill

## 1. Preface

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

For an overview of how to use XMLMill in general please consult in the **docs/** directory:.

- ◆ **apidoc/** -- The JavaDoc concerning the **PDX** api.
- ◆ **userguide.pdf** -- XMLMill user's guide.
- ◆ **dtdguide.pdf** -- An explanation of XMLMill's tag and their attributes.

If you have questions, please do not hesitate to send a mail to [support@xmlmill.com](mailto:support@xmlmill.com).

☞ *This document is completely generated with XMLMill 1.52 using **rnotes152.xml** and **rnotes.xsl**. These files can be found in the directory **samples/docs/rnotes** in the download.*

xmlmill

## 2. Upgrading from previous versions

### 2.1. Upgrading from version 1.51

No changes on .xsl/.mill or Java applications are required.

### 2.2. 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/mill files.
2. Read all release notes [online](#) from your current version to the latest version in correct order (from oldest release notes to newest).
3. Modify your xml/xsl/mill files accordingly to the release notes.

xmlmill

### 3. What's new ?

This release notes document describes all new (or enhanced) functionalities of version 1.52 of XMLMill.

The main changes are:

- ◆ Java API: Dynamically passing top-level (global) stylesheet parameters.
- ◆ Support for PNG images.
- ◆ Bug fixes.

xmlmill

## 4. Java API

This chapter describes the changes of in the PDX API in this release.

☞ *For a full overview of all methods, please visit the [JavaDoc API](#) in the docs/apidocs/ directory in the download.*

The Java API has been extended so top-level (global) xsl stylesheet parameters can be defined at run-time.

### 4.1. Class: PDXTransform

#### 4.1.1. Method: setStylesheetParam(String key, Object expression)

This method allows to define dynamically (at run-time) top-level (global) xsl stylesheet parameters.

The **key** parameter defines the name of the global parameter, the **expression** parameter defines the expression.

The example below passes two parameters from a servlet to a .xsl stylesheet:

```
public class SortOutput extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        String realpath = this.getServletContext().getRealPath("/xmlxsl/");

        PDXTransform transform;

        try
        {
            // Tomcat 4.0.4 server
            File xmlfile = new File(realpath + "/" + request.getParameter("xml"));
            File xslfile = new File(realpath + "/" + request.getParameter("xsl"));

            // Get run-time parameters ❶
            String sortparam = request.getParameter("sortparam");
            String sortorder = request.getParameter("sortorder");

            // Set ContentType
            response.setContentType("application/pdf");

            //Get the output stream
            javax.servlet.ServletOutputStream ostream = response.getOutputStream();

            // Now transform XML and XSL file ...
            transform = new PDXTransform();

            transform.setXMLFile(xmlfile);

            transform.setXSLFile(xslfile);

            // Set transform outputStream
            transform.setOutputStream(ostream);

            // Pass parameter(s) ❷
            transform.setStylesheetParam("sortcolumn", sortparam);
            transform.setStylesheetParam("sortorder", sortorder);

            // and transform. ❸
            transform.transform();

            ...
        }
    }
}
```

❶ The parameters are dynamically defined via the **HttpServletRequest** object.

❷ The parameters are passed to the **PDXTransform** instance using the **setStylesheetParam()** method,

before calling the **transform()** method.

③ The transformation is done.

☞ *To pass more than one parameter, call the **setStylesheetParam** for each parameter separately.*

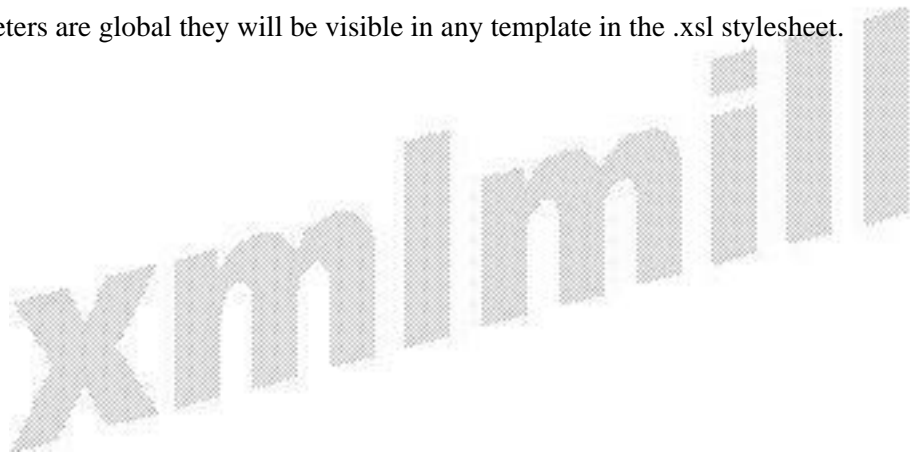
In order to pass the parameter(s) to the .xsl file, the .xsl needs to have defined corresponding top-level (global) parameters:

```
<?xml version="1.0" encoding="UTF-16" ?>
<xsl:stylesheet version="1.0"
  xmlns:ml="http://www.xmlmill.com/XSL/Format "
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xmlmill.com/XSL/Format ../../docs/xsd/xmlmill.xsd"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
>

<xsl:output method="xml" indent="yes" encoding="UTF-16"/>
<xsl:param name="sortcolumn"/> ❶
<xsl:param name="sortorder"/>
...
```

❶ Top-level (global) parameters are defined in the .xsl stylesheet with names corresponding to the **keys** defined in the calling program.

As the parameters are global they will be visible in any template in the .xsl stylesheet.



## 5. PNG Images

### Description:

As of this version XMLMill support PNG Images. Following types are supported:

### Supported color-types:

- ◆ grayscale.
- ◆ grayscale + alpha-channel.
- ◆ color palettes.
- ◆ rgb.
- ◆ rgb + alpha-channel.

### Supported bitdepths (depending on image's color-type):

- ◆ 1 bit.
- ◆ 2 bit.
- ◆ 4 bit.
- ◆ 8 bit.
- ◆ 16 bit.

### Features:

XMLMill can handle PNG files with special features like:

- ◆ interlacing (Adam-7).
- ◆ gamma-support.
- ◆ transparency.

XMLMill has been tested with the [PNG Suite](#) from Willem van Schaik and you should have no problem in using any PNG format.

☞ *In case you have PNG images not supported by XMLMill, please let us know ([support@xmlmill.com](mailto:support@xmlmill.com)) so we can investigate this (and send you a patch).*

### Example:

```

```

## 6. Bugfixes

Following bugfixes were done:

1. The first cell of a table-header did not accept any properties defined by the `<rows>` or `<thead>` tags. This is solved.
2. Wrong layout of a table on a page if a row ends exactly on the end of a page. As a result the following row is placed on the middle of a the same page (should be printed on the next page).
3. Added `<?xml version="1.0" encoding="UTF-16" ?>` in all .xsl examples in the `samples/xmlxsl` directory.
4. Increase of the processing speed of images by using internally a `java.io.BufferedInputStream` stream instead of a regular `java.io.InputStream`.

xmlmill