

XF Rendering Server 2007 - Java Programmers Reference

Copyright© 2002-2006 ECRION Software. All Rights Reserved.

This document is designed to help programmers to develop custom solutions using XF Rendering Server 2007. Please contact Technical Support at support@ecrion.com if you need additional information about this product.

Table of Contents

About XF Rendering Server 2007	3
Product Features	3
Introduction	3
Requirements	3
Feedback and Support	3
Other Resources	4
Java Programmers Reference	5
Summary	5
Class com.ecrion.xf.XFEngine	5
getLog	6
inputFormat	6
encodingOverride	6
baseUrl	6
timeout	6
outputFormat	6
setProperty	7
render	9
renderUrl	9
print	9
printUrl	10
Class com.ecrion.xf.XFDocument	10
printTo	11
exportTo	11
getPageCount	11
getPageWidth	11
getPageHeight	12
load	12
loadUrl	12
Class com.ecrion.xf.XFMergeContext	12
outputFormat	13
setOutput	13

Last updated: August 2006

Important Notice: This document and the information within is furnished "as is" and is subject to change without notice. In no event shall the author be liable for any damages whatsoever (including, without limitation, damages for loss of business profits, business interruption, loss of business information, or any other pecuniary loss) arising out of the use of or inability to use this product, even if the author has been advised of the possibility of such damages.

ECRION

This document has been generated using XF Rendering Server 2007. For the latest version, visit [Technical Resources](#) section on our web site.

About XF Rendering Server 2007

XF Rendering Server 2007 is a highly scalable, enterprise class rendering product. It can be used to automate the creation of electronic documents like technical manuals, brochures, proposals, business reports containing charts and graphs, by dynamically generating them from XML.

XF Rendering Server 2007 supports two major industry standards: XSL-FO (Extensible Style Language Formatting Objects) describing how an XML document should be formatted for a variety of media as well as SVG (Scalable Vector Graphics) used to describe two-dimensional vector and mixed vector/raster graphics in XML.

In addition high quality, all-purpose charts can be generated directly from XML using **xChart** XML language. More information can be found in [xChart 1.0 Language Reference](#).

Product Features

- Supports XSL-FO, SVG, WordML, xChart as input.
- Produces PDF, Postscript, HTML, GIF, JPEG, PNG, BMP, TIFF, AFP and other formats.
- Supports TrueType and Postscript Type 1 font embedding.
- Scalable server architecture that can run across multiple CPUs, meeting the high-performance needs of your applications.
- Available in 32 bit and 64 bit editions.
- Is accessible from a multitude of development environments: C++, VB, ASP, Java, Java.
- Includes XF Designer 2004 XSL-FO authoring tool.

Introduction

This reference describes the XF Rendering Server 2007 programming interfaces available in the Java environment.

This SDK is not intended to be an in-depth tutorial on XSL-FO, SVG or on any related standards, and does not replace the World Wide Web Consortium (W3C) specifications for these standards. The Other Resources section below provides links to these standards on the Internet.

This SDK is not intended to demonstrate every possible use of XF, and does not document every known problem with XF. See Help and Support, below, for links to support resources.

Among the core services, XF Rendering Server 2007 provides, is developer support for the following:

- Programatically convert XSL-FO, SVG and xChart to PDF as well as other output formats using com.ecrion.xf package.
- Generate PDF documents directly in Web Server's output stream.

Requirements

com.ecrion.xf package can be used in the Microsoft Windows® 2000, Windows XP and Windows Server 2003 environments.

- Minimum Intel Pentium III, AMD Athlon 500 MHz or better. Intel Pentium IV 2.4 GHz recommended for development computers, dual XEON 3.0 GHz for production servers.
- Minimum 128 Mb RAM, 512 Mb recommended for development computers, 1Gb for productions servers.

Feedback and Support

Send error reports, feature requests, and comments about the XF documentation or samples directly to the [Technical Support team](#).

Further information about support options can be found on the [Ecrion Web site](#).

Other Resources

[XML Recommendation](#)

[XSLT Recommendation](#)

[XSL-FO Recommendation](#)

[SVG Recommendation](#)

[XF Rendering Server Users Guide](#)

[xChart 1.0 Language Reference](#)

[XF Tutorial and QuickStarts](#)

[XF-Java Tutorial and QuickStarts](#)

Java Programmers Reference

Summary

The following table lists the core objects:

Class	Description
XFEngine	Class representing a single instance of a rendering engine. Use-it to render XSL-FO, SVG or xChart documents in one single step.
XFDocument	Class representing a single instance of a document. Use-it to manage XSL-FO, SVG or xChart documents.
XFMergeContext	Class representing the result of multiple rendering operations. Use-it in conjunction with XFDocument to generate on large output from multiple input documents.
XFException	The exceptions that could occur using XFDocument, XEngine or XFMergeContext class.
XFRenderer	Old class that unites the functionality of XFEngine and XFDocument. This class is deprecated. Use instead XFDocument and XFEngine.
XFRendererException	The exceptions that could occur using XFRenderer class. This class is deprecated.

Class [com.ecrion.xf.XFEngine](#)

XFEngine represents a single instance of a renderer. Use-it to render both XSL-FO, SVG or xChart documents into PDF or a raster format.

Accessor methods

This methods are used to access or modify internal engine properties.

Name	Description
getLog	The list of events occurred during rendering.
setInputFormat	Sets the input format.
getInputFormat	Gets the input format.
setEncodingOverride	Overwrites input's encoding.
setBaseUrl	Sets the base URL to be used when computing relative paths.
setTimeout	Sets timeout period assigned to the rendering job, in seconds.
getTimeout	Gets timeout period assigned to the rendering job, in seconds.
setOutputFormat	Sets the output's format.
getOutputFormat	Gets the output's format.

Methods

Name	Description
setProperty	Set various rendering parameters.
render	Render a document using the supplied input. Overloaded.
renderUrl	Render the document from the specified location. Overloaded.
print	Prints the document using the supplied input. Overloaded.

Name	Description
printUrl	Prints the document from the specified location.

getLog

The list of events occurred during rendering.

inputFormat

Read/write accessors `getInputFormat/setInputFormat` `get/set` input's format engine property. Can take one of the following values:

XIF_Auto	Specifies that the input format should be auto-detected. This is the default value for the input format.
XIF_Svg	The XML input is in SVG (Scalable Vector Graphics) format.
XIF_XslFo	Input is an XSL-FO document.
XIF_XChart	Input is an xChart document.
XIF_WordML	Input is Word XML document.
XIF_HTML	Input is HTML text.

encodingOverride

Write-only property that can be used to override input's encoding. This property is set through `setEncoding` accessor. If this property is not set, the engine will use the encoding specified in the `<xml>` declaration, or if not present, it will try to autodetect-it.

baseUrl

Write-only String property that represents the URL of the current XML document. This property is set through `setBaseUrl` accessor. This property is very useful when the input is provided as String or Stream.

The renderer uses this base URL to resolve relative links. To resolve these links, the reader uses the base URL in much the same way that an Internet browser uses a Web page's URL to resolve relative links contained on that page. The base URL may be a complete URL. However, the reader only uses that portion of the URL up to and including the final backslash when resolving links. For instance, a base URL for a document is "C:\Documents\XSL-FO\test.fo". The reader would then only use "C:\Documents\XSL-FO\" when resolving links. The same result would be obtained if you specify "C:\Documents\XSL-FO\".

As a final note the base url can also a http or https type of URL. In this case, if you specify only the folder of the current document you have to include a final backslash.

timeout

Read/Write integer property representing the maximum amount of time (in seconds) that the rendering should take. This property can be set/get through `setTimeout/getTimeout` accessors. This property can be used to prevent the server being blocked by large jobs.

The initial value is 0, that is, the rendering will never be interrupted.

outputFormat

Read/write property containing input's format. This property can be set/get through setTimeout/getTimeout accessors. Can take one of the following values:

XFEngine.XOF_Bitmap	Windows Bitmap
XFEngine.XOF_Gif	GIF89
XFEngine.XOF_Png	Portable Network Graphics
XFEngine.XOF_Jpeg	JPEG
XFEngine.XOF_Tiff	TIFF.
XFEngine.XOF_PDF	Adobes' PDF (Portable Document Format).
XFEngine.XOF_Html	Specifies that the output format must be HTML.
XFEngine.XOF_SVG	Scalable Vector Graphics.
XFEngine.XOF_PostfScript	Specifies that the output format must be PostScript. You should always render to a file instead of a stream because PostScript files can be very large. This will preserve server's memory and result in a more scalable application.
XOF_Default	If the output is a file, the output format can be determined from the output file's extension. If the extension is not recognized or if the output's destination is not a file, an exception is thrown, that is, the output format must be specified explicitly. This is the default value.

setProperty

Set various rendering parameters.

```
void setProperty(String propertyName, String newVal);
void setProperty(String propertyName, int newVal);
```

Parameters

propertyName

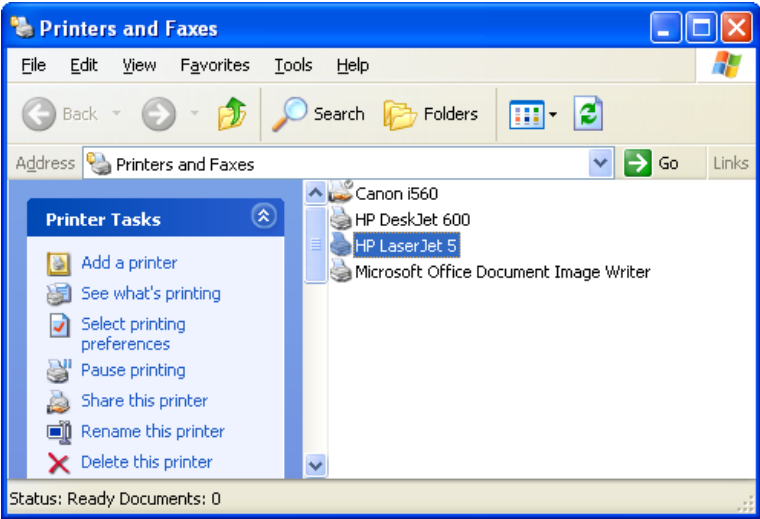
The name of the property you wish to change. For a list of valid property names see "Remarks" section.

newVal

The new value for this property.

Remarks

Name	Description
ColorDepth	The number of bits used to describe the color of a single pixel, when the output is a raster image format (GIF, JPEG, etc). The value can be 1 (Monochrome), 8 (256 Colors) or 24 (True Color, also the default value).
XResolution YResolution	Integer property describing the output image's resolution in DPI (Dots Per Inch). The default value for both resolutions is 96.

Name	Description
Compression	<p>When output format is <i>TIFF</i>, this can take the following values: "NONE" (the default), "RLE", "LZW", "CCITT3", "CCITT4".</p> <p>When output format is <i>JPEG</i>, this is an integer value between 0 and 100.</p>
JobPrinterDriver	<p>When output format is PCL, you can specify which printer driver must be used to generate PCL. For example, considering the following setup:</p>  <p>You can specify "HP LaserJet 5" or "HP DeskJet 600" for this property.</p> <p>Note: A Windows printer driver can be installed without actually having a printer connected. This property performs an override of the global settings configured though XF Rendering Server 2007 Management Console. It will affect only the current rendering job.</p>
Antialiasing	<p>When output format is a raster format (GIF, JPEG, PNG, etc), you can specify the process used to render text.</p> <p>It can take one of the following values:</p> <ul style="list-style-type: none"> • <i>"SystemDefault"</i> - Specifies that a character is drawn using the currently selected system font smoothing mode. This is the default value. • <i>"SingleBitPerPixelGridFit"</i> - Specifies that a character is drawn using its glyph bitmap and hinting to improve character appearance on stems and curvature. • <i>"SingleBitPerPixel"</i> - Specifies that a character is drawn using its glyph bitmap and no hinting. • <i>"AntiAliasGridFit"</i> - Specifies that a character is drawn using its antialiased glyph bitmap and hinting. This results in much better quality due to antialiasing. • <i>"AntiAlias"</i> - Specifies that a character is drawn using its antialiased glyph bitmap and no hinting. Stem width differences may be noticeable because hinting is turned off.

Name	Description
	<ul style="list-style-type: none"> • <i>"ClearTypeGridFit"</i> - Specifies that a character is drawn using its glyph ClearType bitmap and hinting. <p>Windows XP and Windows .NET Server only: ClearType rendering is supported only on Windows XP and Windows .NET Server.</p>

render

Render a document using the supplied input.

```
public void render(String xml, String outputFileName)
public void render(String xml, OutputStream outputStream)
public void render(Stream inputStream, String outputFileName)
public void render(Stream inputStream, OutputStream outputStream)
```

Parameters

xml

A valid XML document containing text to render.

inputStream

A stream containing XML text to render.

outputFileName

The file path where the renderer should place the output.

outputFileStream

The destination stream to for rendering's output.

renderUrl

Render the document from the specified location.

```
public void renderUrl(String url, String outputFileName)
public void renderUrl(String url, OutputStream outputStream)
```

Parameters

url

Path to a valid XML document containing text to render.

outputFileName

The file path where the renderer should place the output.

outputFileStream

The destination stream to for rendering's output.

print

Prints a document using the supplied input.

```
public void print(string xml, string printerName, PrintMode mode)
public void print(InputStream inputStream, string printerName, PrintMode mode)
```

Parameters

xml

A valid XML document containing text to print.

InputStream

A stream containing XML text to print.

printerName

The name of the printer that should print the document.

PrintMode

The mode in which printing would be made. It supports three modes GDI, PCL, POSTSCRIPT

printUrl

Prints the document from the specified location.

```
public void printUrl(string url, string printerName, PrintMode mode)
```

Parameters

url

A valid location containing text to print.

printerName

The name of the printer that should print the document.

PrintMode

The mode in which printing would be made. It supports three modes GDI, PCL, POSTSCRIPT

Class com.ecrion.xf.XFDocument

XFDocument represents a single instance of a document. Use-it to render both XSL-FO, SVG or xChart documents into PDF or a raster format.

Accessor methods

This methods are used to access or modify internal engine properties.

Name	Description
getLog	See the documentation for XFEngine class.
setInputFormat	See the documentation for XFEngine class.
getInputFormat	See the documentation for XFEngine class.
setEncodingOverride	See the documentation for XFEngine class.
setBaseUrl	See the documentation for XFEngine class.
setTimeout	See the documentation for XFEngine class.
getTimeout	See the documentation for XFEngine class.
setOutputFormat	See the documentation for XFEngine class.
getOutputFormat	See the documentation for XFEngine class.

Methods

Name	Description
setProperty	See the documentation for XFEngine.
printTo	Prints a document previously loaded.
exportTo	Exports the document previously loaded. Overloaded.
getPageCount	Returns the number of pages for a loaded document.

Name	Description
getPageWidth	Returns the width of a page in pixels.
getPageHeight	Returns the height of a page in pixels.
load	Loads the XML document contained in the given context. Overloaded.
loadUrl	Loads the XML document specified by the given url.

printTo

Prints a document using previously loaded with load or loadUrl.

```
public void printTo(string printerName, PrintMode mode)
```

Parameters

printerName

The name of the printer that should print the document.

PrintMode

The mode in witch printing would be made. It supports three modes GDI, PCL, POSTSCRIPT

exportTo

Exports the document previously loaded. Output format can be recognized from the output file's extension when the export context is a file. When the export context is a stream the OutputFormat must be set before calling this function.

```
public void exportTo(String url)
public void exportTo(OutputStream outputStream)
public void exportTo(XFMergeContext ctx)
```

Parameters

outputFileName

Name of the file to receive rendering output.

outPutStream

Stream to receive rendering output.

ctx

A merge context that will receive the output.

getPageCount

Returns the number of pages for a loaded document.

```
public int getPageCount()
```

getPageWidth

Returns the width of a page in pixels.

```
public double getPageWidth(int idx)
```

Parameters

idx

Page index

getPageHeight

Returns the height of a page in pixels.

```
public double getPageHeight(int idx)
```

Parameters

idx

Page index

load

Loads the XML document contained in the given context.

```
public void load(string xml)
public void load(InputStream inputStream)
```

Parameters

xml

Load the XML document contained in the given string.

inputStream

Load the XML document contained in the given stream.

loadUrl

Load the XML document specified by the given url

```
public void loadUrl(string url)
```

Parameters

url

Path to document to be loaded.

[Class com.ecrion.xf.XFMergeContext](#)

XFMergeContext can be used to render multiple input documents into one output file or stream. This feature can very useful for printing jobs which require all inputs to be sent to a printer into one large file Postscript file. It can also be used for archiving purposes.

The following code fragment exemplifies this feature:

```
import com.ecrion.xf.*;

static void mergeOutputs(string[] inputFiles, string outputFile, int fmt)
{
    XFMergeContext ctx = new XFMergeContext();
    XFDocument doc = new XFDocument();
    try
```

```

    {
        ctx.setOutputFormat = fmt;
        ctx.setOutput(outputFile);
        for (int i=0; i<inputFiles.length; i++)
        {
            doc.loadUrl(inputFiles[i]);
            doc.exportTo(ctx);
        }
    }
    catch (Exception e)
    {
        System.out.println("An error has occurred : "+e);
    }
    finally
    {
        ctx.dispose();//Both merge context and document instances must be
        doc.dispose();//disposed properly
    }
}

```

You can also use a `OutputStream` object as parameter to `setOutput` method.

Please note that the merge context must be closed gracefully in order to obtain a valid output file.

Accessor Methods

Name	Description
setOutputFormat	Sets the output's format.
getOutputFormat	Gets the output's format.

Methods

Name	Description
setOutput	Sets the output file or stream.

outputFormat

Sets or gets the output format of the merged output.

setOutput

Sets the output file or stream

```

void setOutput(string url)
void setOutput(OutputStream outputStream)

```

Parameters

url

The complete path of the file that will hold the output.

outputStream

The `OutputStream` object that will hold the output.