Apache™ FOP Development: RTFLib (jfor)

Version 1298724

Table of contents

1 General Information 2
1.1 Introduction 2
1.2 History 2
1.3 Status 2
2 User Documentation 2
2.1 Overview 2
2.2 Document Structure 3
2.3 Attributes 3

1 General Information

1.1 Introduction

The RTFLib package is an open-source, *independent* package suitable for writing RTF files in a java environment. By *independent* we mean:

- Although it is used by Apache™ FOP to generate FOP's RTF output, it is not dependent on FOP for any of its functionality.
- It does not require FOP as a front-end, nor does it even require XSL-FO for input. It essentially exposes an API of relatively high-level RTF construction routines to a host program. This means it can be used anywhere RTF output is required and java is available.

The FOP development team intends to keep the RTFLib package independent so that it can be used for other purposes.

1.2 History

RTFLib was originally developed by Bertrand Delacrétaz and the jfor team. jfor was written under an Apache-style license, and the jfor team contributed the code to the Apache Software Foundation in June, 2003. RTFLib is a subset of the original jfor project, which also includes an XSL-FO parsing mechanism for a complete XSL-FO to RTF solution.

1.3 Status

Although FOP's implementation of the RTFLib package is very incomplete, the RTFLib package itself is relatively mature. RTFLib is only available in the trunk line of FOP development.

Warning:
This documentation is a work in progress. If you see errors or omissions, please report them to the fop-dev mailing list.

2 User Documentation

2.1 Overview

Perhaps the easiest way to see how to use RTFLib is by looking at an example. A set of test documents is part of the package, and can be viewed online. A quick look at the Abstract TestDocument class, and one of the Concrete subclasses, SimpleDocument will provide the basics of how to use the package.

There are two basic concepts you will need to understand in order to use RTFLib:

- Documents are created by filling bigger containers with successively smaller containers, and eventually with content.
- Attributes may be set for each container or content as it is created

RTFLib handles the process of converting to and writing the RTF content as the document is created. All you need to do is flush the document at the end to make sure that the last pieces get written.

2.2 Document Structure

Warning:
This section is very incomplete.

The documentation in this section is intended to provide a high-level view of the process of building an RTF document. For more detailed API documentation of the various methods, be sure to consult the Javadocs for RTFLib.

The following table summarizes the various containers that can be created:

Name	Class.Method where created	Attribute Set(s)	Valid children
Document Area	RtfFile.startDocumentArea()	Information Group, Document Formatting	Section
Section	RtfDocumentArea.newSection	Section Formatting	Paragraph, ParagraphKeepTogether, Image, List, Before (Page Heading), After (Page Footer), JforCmd
Paragraph	RtfSection.newParagraph()	Paragraph Formatting, Character Formatting	Text
ParagraphKeepTogether	RtfSection.newParagraphKeep	.	.
Image	RtfSection.newImage	.	.
Table	RtfSection.newTable	.	.
List	RtfSection.newList	.	.
Before (Page Heading)	RtfSection.newBefore	.	.
After (Page Footer)	RtfSection.newAfter	.	.
JforCmd	RtfSection.newJforCmd	.	.
Text	RtfParagraph.newText()	Character Formatting	N/A

2.3 Attributes

Warning:
This section is very incomplete.

Attributes can be set for each container and piece of content in the document. The general approach is to build an RtfAttributes object containing the various attributes, then pass that RtfAttributes object to the method that creates the new container or content. See the Javadoc API documentation for RtfAttributes for details on the syntax for creating an RtfAttributes object. The following information lists the various attributes that can be set for each type of container.

2.3.1 Information Group

These attributes are set when creating a Document.

2.3.2 Document Formatting

These attributes are set when creating a Document.

2.3.3 Section Formatting

These attributes are set when creating a Section.

2.3.4 Paragraph Formatting

These attributes are set when creating a Paragraph.

Description	Attribute Name	Attribute Value	RTF Control Word
Alignment		.	
Indentation		.	
Borders		.	
.			
Align Left	RtfText.ALIGN_LEFT	N/A (boolean)	\ql
Align Right	RtfText.ALIGN_RIGHT	N/A (boolean)	\qr
Align Centered	RtfText.ALIGN_CENTER	N/A (boolean)	\qc
Align Justified	RtfText.ALIGN_JUSTIFIED	N/A (boolean)	\qj
Align Distributed	RtfText.ALIGN_DISTRIBUT	N/A (boolean)	\qd
Kashida justification	not implemented	0-20 (integer)	\qkN
Thai Distributed justification	not implemented	N/A (boolean)	\qt
Left indent body	RtfText.LEFT_INDENT_BOI	(int) "hundredths of a character unit" (?)	\li
Left indent first	RtfText.LEFT_INDENT_FIR	(int) "hundredths of a character unit" (?)	\fi
Bottom single border	RtfText.BDR_BOTTOM_SIN	Boolean?	brdrb\\brsp40\\brdrs
Bottom double border	RtfText.BDR_BOTTOM_DO	Boolean?	brdrb\\brsp40\\brdrdb
Bottom embossed border	RtfText.BDR_BOTTOM_EM	Boolean?	brdrb\\brsp40\\brdremboss
bottom dotted border	RtfText.BDR_BOTTOM_DO	Boolean?	brdrb\\brsp40\\brdrdot
bottom dashed border	RtfText.BDR_BOTTOM_DA	Boolean?	brdrb\\brsp40\\brdrdash

2.3.5 Character Formatting

These attributes are set when creating a Paragraph, or Text.

Description	Attribute Name	Attribute Value	RTF Control Word
Bold	RtfText.ATTR_BOLD	N/A (boolean)	\b
Italic	RtfText.ATTR_ITALIC	N/A (boolean)	\i
Underline	RtfText.ATTR_UNDERLINE	N/A (boolean), or (int) 0 to turn underlining off	\ul
Font Size	RtfText.ATTR_FONT_SIZE	(int) font size in half-points	\fs
Font Family	RtfText.ATTR_FONT_FAMI	(int) entry in document font-table	\f
Font Color	RtfText.ATTR_FONT_COLC	(int) entry in document color-table	\cf
Background Color	RtfText.ATTR_BACKGROU	(int) entry in document color-table	\chcbpat