

# Web Enablement Solutions Suite

# Line2PDF Plus User's Guide

Overview

- 1

Running Java Version of Line2PDF Plus

2

**Version 1.3** 

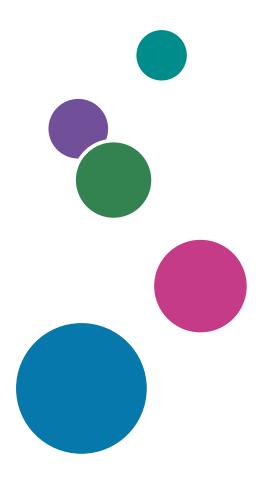
Client/Server Model

3

For information not in this manual, refer to the Help System in your product.







# **TABLE OF CONTENTS**

# Introduction

Important	2
Cautions regarding this guide	2
Publications for the Web Enablement Solutions Suite	2
Symbols	3
Abbreviations	4
Trademarks	4
1 Overview	
Benefits	7
Requirements	7
Limitations	8
2 Running Java Version of Line2PDF Plus	
Java Command Line	
Line2PDF Plus Command Notation	9
Java Line2PDF Plus Command Options	10
Running Java Line2PDF Plus Server	14
Running Java Agent on Client Machine	15
3 Client/Server Model	
Client Native C/C++ Program	
Client and Server on the Same Machine	18
Client and Server on Different Machines	18
Enabling Debug Mode for Server and Agent	19

# Introduction

#### **Important**

To the maximum extent permitted by applicable laws, in no event will the manufacturer be liable for any damages whatsoever arising out of failures of this product, losses of documents or data, or the use or non-use of this product and operation manuals provided with it.

Make sure that you always copy or have backups of important documents or data. Documents or data might be erased due to your operational errors or malfunctions of the software. Also, you are responsible for taking protective measures against computer viruses, worms, and other harmful software.

In no event will the manufacturer be responsible for any documents created by you using this product or any results from the data executed by you.

#### Cautions regarding this guide

- Some illustrations or explanations in this guide could differ from your product due to improvement or change in the product.
- The contents of this document are subject to change without notice.
- No part of this document may be duplicated, replicated, reproduced in any form, modified, or quoted without prior consent of the supplier.

#### **Publications for the Web Enablement Solutions Suite**

This section provides a list of all the publications for the Web Enablement Solutions Suite transforms.

## Instruction manuals

#### AFP2PDF PLUS

For information about AFP2PDF PLUS, see these documents:

- AFP2PDF Plus User Guide 1.300
- AFP2PDF Plus Quick Start Guide
- AFP2PDF Plus Setup Guide 1.300
- AFP2PDF Plus Release Notes 1.300
- AFP2PDF Plus Summary of Updates

#### AFP Visual Environment

For information about AFP Visual Environment, see these documents:

- AFP Visual Environment User Guide 6.6
- AFP Visual Environment Release Notes 6.6
- AFP Visual Environment 6.6 Limitations List

#### AFPMerge

For information about AFPMerge, see these documents:

- AFPMerge User Guide 3.3
- AFPMerge Release Notes 3.3
- AFPMerge 3.3 Limitations List

#### Line2PDF Plus

For information about Line2PDF Plus, see these documents:

- Line2PDF Plus User's Guide 1.3
- Line2PDF Release Notes 1.3
- Line2PDF Plus 1.3 Limitations List

#### PCL2PDF

For information about PCL2PDF, see these documents:

- PCL2PDF 1.3 User's Guide
- PCL2PDF 1.3 Release Notes
- PCL2PDF 1.3 Limitations List

#### PS2PDF

For information about PS2PDF, see these documents:

- PS2PDF 1.3 User's Guide
- PS2PDF 1.3 Release Notes
- PS2PDF 1.3 Limitations List

#### Tiff2PDF

For information about Tiff2PDF, see these documents:

- Tiff2PDF Plus User Guide 1.3
- Tiff2PDF Plus Release Notes 1.3
- Tiff2PDF Plus 1.3 Limitations List

## **Symbols**

The following symbols are used in this manual to help you to identify content quickly.



• This symbol indicates points to pay attention to when using the product. Be sure to read these explanations.



This symbol indicates helpful supplementary information that is not essential to completing a task.

**Bold type** indicates the names of commands and parameters.

**Bold underline Underlined bold type** indicates the default value.

Italic type indicates variables that you must replace with your own

information.

Monospace type indicates computer input and output and file names.

[] Square brackets indicate that a value is optional.

A vertical bar indicates a choice between values.

··· An ellipsis indicates that a series can continue.

#### **Abbreviations**

#### **ANSI**

American National Standards Institute

#### **API**

**Application Programming Interface** 

#### **ASCII**

American Standard Code For Information Interchange

#### CC

Carriage Control

#### **CMOD**

Content Manager OnDemand

#### **DBCS**

Double Byte Character Set

#### **EBCDIC**

Extended binary-coded decimal interchange code

#### **PDF**

Portable Document Format

#### **TRC**

Table Reference Character

#### **Trademarks**

These terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- IBM
- z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo, PDF, and the PDF logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Oracle, Java are registered trademarks of Oracle Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

The proper names of the Windows operating systems are as follows:

- Windows 10:
  - Microsoft Windows 10 Pro
  - Microsoft Windows 10 Enterprise
- Windows Server 2016:
  - Microsoft Windows Server 2016 Standard

# 1. Overview

- Benefits
- Requirements
- Limitations

The Line2PDF Plus Transform converts simple line data into Adobe Acrobat Portable Document Format (PDF) files. The text is quickly transformed and displayed on the screen.

It handles both ASCII and EBCDIC and ANSI and machine carriage control. You interact with Line2PDF Plus through a command line or JAVA API interface.

The Line2PDF Plus transform lets you:

- Operate on any system that supports JAVA Version 1.8 or later.
- Integrate multiple servers with little or no client workstation modification. The Line2PDF Plus transform runs on your Web server or other back-end application server.
- View documents with the same fidelity as if they were printed. If the Adobe Acrobat plug-in is
  installed with a Web browser, you can view and print these documents within the browser
  application.
- Fully integrate with the IBM Content Manager OnDemand (CMOD) Web Enablement Kit and IBM Content Navigator.

# **Benefits**

The Line2PDF Plus Transform gives you these added benefits:

- Client / Server implementation.
- Greatly improved transformation speed over Line2PDF.
- Larger than 2 GB input and output file support.
- Reduce costs associated with printing and mailing by delivering documents electronically.
- Quickly retrieve your information within multi-page documents using Adobe Acrobat search and navigation features.
- Print Line2PDF Plus Transform documents on any local printer using the print function in Adobe Acrobat.

# Requirements

#### **Transform client requirements:**

IBM AIX 7.1 or later

IBM z/OS UNIX System Services V1.13 or later

Microsoft Windows 10 Pro, Enterprise or later

Microsoft Windows Server 2016 Std or later

Linux Kernel 2.6.18 or later (x86)

Linux Kernel 2.6.09 or later (IBM System z)

#### Transform server requirements:

JAVA V1.8 or later

# End-user client requirements:

Adobe Acrobat, Acrobat Reader, or Acrobat Plug-In 9.0 or later

# Limitations

The current limitations with the Line2PDF PlusTransform include:

- Input text data stream must be mappable to code page 1252. Double Byte Character Set (DBCS) is not supported.
- TRC characters are not supported.

# 2. Running Java Version of Line2PDF Plus

- Java Command Line
- Running Java Line2PDF Plus Server
- Running Java Agent on Client Machine

To run Java on Line2PDF Plus use one of these options:

- Java command line.
- Client/Server Model.

# **Java Command Line**

To transform a Line data into a PDF:

- 1. Run Java command. This requires a javaline2pdf.jar file.
- 2. Set CLASSPATH to include the full path of the jar file.
  - For Windows:

## set CLASSPATH=<directory>\javaline2pdf.jar;%CLASSPATH%

For example:

set CLASSPATH=C:\Line2pdf\javaline2pdf.jar;%CLASSPATH%

• For other operating systems:

# export CLASSPATH=<directory>/javaline2pdf.jar;\$CLASSPATH

For example:

export CLASSPATH=/Line2pdf/javaline2pdf.jar; \$CLASSPATH



Setting CLASSPATH should be on a single line.

## **Line2PDF Plus Command Notation**

Command syntax notation uses symbols to show specific conditions:

Bar

П

**Brackets** 

[]

Underlining

. .

**Ellipsis** 

•••

These symbols have the following meanings when used in the publications and in the online information:

- A vertical bar, |, between values means that you can only enter one of the values with the command.
- Brackets, [], around values indicate that they are optional.
- An ellipsis, ..., means that you can supply more than one occurrence of a keyword or value with the command.
- Underlined text identifies the default fixed value the transform uses if you do not specify a value.

# Java Line2PDF Plus Command Options

#### Command line syntax

```
java line2pdf [options] input-file output-file
```

For example:

- For Windows: java line2pdf c:\linedata.txt c:\linedata.pdf
- For other operating systems: java line2pdf /linedata.txt /linedata.pdf

or

```
java line2pdf [options] output-file < input-file</pre>
```

For example:

- For Windows: java line2pdf c:\linedata.pdf < c:\linedata.txt
- For other operating systems: java line2pdf /linedata.pdf < /linedata.txt

#### Options:

-g

Specifies an input configuration file that defines channels.

#### Usage note:

 The first line of the channel control file is [CHANNEL INFO], followed by 12 entries consisting of numbers that specify the landing line number where "Skip to channel n" should print. Default values:

```
[CHANNEL INFO]
1
7
13
19
25
31
37
43
63
49
```

-r

Replaces existing output file.



• The Line2PDF Plus Transform does not overwrite an existing output file by default.

To determine the outlook of the PDF file:

#### -wNNN

Specifies the width of the page to NNN. The units of this parameter are 72 units per inch.

Default value: 612.

#### -hNNN

Specifies the height of the page to NNN. The units of this parameter are 72 units per inch.

Default value: 792.

#### —n nnn

Specifies the lines per page to nnn.

#### -s n.n

Specifies the line spacing.

Default value: 1.0.

#### -p nn

Specifies the font size.

Default value: 12.

#### —f fn

Specifies a font from the following list:

- Courier
- Courier-Bold
- Courier-Oblique
- Courier-BoldOblique
- Helvetica
- Helvetica-Bold
- Helvetica-Oblique
- Helvetica-BoldOblique
- Symbol
- Times-Roman
- Times-Bold
- Times-Italic
- Times-BoldItalic
- ZapfDingbats

Default value: Courier.

#### -b nn

Specifies the border size. The units of this parameter are 72 units per inch.

Default value: 20.

#### -m lm tm bm

Specifies the page margins: **Im** for left margin, **tm** for top margin and **bm** for bottom margin. The units of this parameter are 72 units per inch.

To determine the type, format and line break for the line data:

#### -x nnn

Selects the type of carriage control.

- 0: no carriage control character
- 2: ANSI
- 3: ANSI with TRC
- 4: Machine CC
- 5: Machine CC with TRC

#### -I nnn

Specifies a fix line length, in characters or bytes, where nnn=one of [ 0 | positive number[,[char| byte]] | record ]

- 1. If nnn=0 (default), new line characters is applied as line break.
- 2. If nnn>0, each line has nnn chars or bytes. Default value is specified in chars.

For example:

- -1 80.char
- -1 80, byte
- 3. If nnn=[record | record+ | record-], variable length record apply:

The first 2 bytes of the record indicate the record length.

- If **nnn** specifies the record or record+, the record length includes the length of the first 2 bytes.
- If nnn specifies the record-, the record length excludes the length of the first 2 bytes.

#### —a nnn

Selects a code page or data format, where nnn=one of [ASCII|EBDIC|UTF8|UTF16|codepage number].

Default: ASCII.

- If ASCII is specified, 437 is used for the code page.
- If EBCDEC is specified, 500 is used for the code page.

#### -lm nnn

Specifies the maximum line length to convert.

Other options:

#### -user pass

Specifies the user password.



• For an encrypted output, —user pass and —pw:psca pass options must be used together with bcprov-jarfileversion.jar. Make sure to add the full path of the jar file to CLASSPATH. See Java Command Line, p. 9 for more information on setting CLASSPATH.

#### -pw:psca pass

Protects data with password and security flags.

a

Adds or modifies text annotations and interactive form fields.

C

Modifies the document contents.

р

Prints the document.

S

Copies the text and graphics from the document.

## -bglogo fn

Applies the logo/preprint form, where **fn** is the PDF file name.



• If Client and Server are running on different machines, the **fn** used with the **-bglogo** option must be a path or a file name, or both, located on the Server machine, not on the Client machine.

#### -bglogo +"matrix" fn

Applies matrix to logo/preprint form (6 numbers).

```
tm1 tm2 tm3 tm4 tm5 tm6
```

Specifies the linear transformation matrix values that define the origin, rotation, or scale of the appended PDF file. For example:

#### 1001 ox oy

Specifies origin changes, where:

ОX

Specifies the distance to translate the horizontal dimension.

o,y

Specifies the distance to translate the vertical dimension.

#### sx 0 0 sy 0 0

Specifies scaling changes, where:

sx

Specifies the scale factor (1.0-100%) for the horizontal direction.

Sy

Specifies the scale factor (1.0-100%) for the vertical direction.

```
\cos \theta \sin \theta - \sin \theta \cos \theta 00
```

Specifies rotation changes by an angle  $\theta$  counterclockwise.

The origin for this operation is the bottom left-hand corner of the page.

# -bglogo: range fn

Specifies individual pages or a range of pages to apply the logo/preprint form.

The range value can include individual page numbers or a range of pages.

- To separate pages and page ranges, use commas.
- To specify a range of pages, use a hyphen between the first and last page. For example: 1,150,212-240,332.

#### -bglogo: range +"matrix" fn

Specifies individual pages or a range of pages to apply the matrix to logo/preprint form.

For more information, see **—bglogo +"matrix" fn** and **—bglogo: range fn**.

#### Return code

O: if no error.

Other numbers: if any error occurs.

# **Running Java Line2PDF Plus Server**

Set CLASSPATH to include javaline2pdf.jar.

# **Command line syntax**

```
java Line2PdfServer [-start] | [-stop] [-p socket_port] [-r rmi_port] [-t temp_dir]
```

For example:

For Windows:

```
java Line2PdfServer -start
or
java Line2PdfServer -start -p 8500 -t c:\temp
```

For other operating systems:

```
java Line2PdfServer -start
or
java Line2PdfServer -start -p 8500 -t /temp
```

Values:

#### -start

Starts the server.

#### -stop

Stops the server.

## -p socket\_port (optional)

Specifies the socket port number for local client.

Default value:8500.

#### —r rmi\_port (optional)

Specifies the RMI port number for remote Line2PdfServer.

Default value: 8501.

#### —t temp\_dir (optional)

Specifies the temporary directory name.

Default: <currentDir>/temp.

## **Examples for running Server**

## Example 1

1. To start Line2PdfServer set these values:

#### Client port:

Default **8500**.

# **RMI** port:

Default 8501.

- 2. Enter the command line: java Line2PdfServer -start.
- 3. To stop Line2PdfServer enter: java Line2PdfServer stop.

#### Example 2

1. To start Line2Pdf Server set these values:

Client port: 7200.

RMI port: 7201.

Tempdir: /mydir/tmp.

- 2. Enter the command line: java Line2PdfServer -start -p 7200 -r 7201 -t /mydir/tmp.
- 3. To stop Line2PdfServer enter: java Line2PdfServer -stop -p 7200.

# **Running Java Agent on Client Machine**

Set CLASSPATH to include:

javaline2pdf.jar

# **Command line syntax**

#### To start Line2PdfAgent:

Java Line2PdfAgent -s server\_host\_name [-p client\_port] [-sp server\_rmi\_port]

#### To stop Line2PdfAgent:

Java Line2PdfAgent -stop [-p client\_port]

For example: java Line2PdfAgent -p 8500 -s C7YZV32.us.ricoh.ds -sp 8501

Values:

# -p client\_port optional

The port number line2pdf is connected to.

Default value: 8500.

-s server\_host\_name

The server host name where Line2PdfServer is running.

-sp server\_rmi\_port (optional)

The server port number Line2PdfAgent is connected to.

Default value: 8501.

-stop

Stop the Line2PdfAgent.

## **Examples for running Agent on client machine**

#### Example 1

1. To start Line2Pdf Agent set these values:

**Client port:** 

Default 8500 for C/C++ client.

Remote host where Line2PdfServer is running: server1.

**RMI** port:

Default 8501.

- 2. Enter the command line: java Line2PdfAgent —s server1.
- 3. To stop Line2PdfAgent enter: java Line2PdfAgent —stop.

#### Example 2

1. To start Line2PdfAgent set these values:

Client port: 9500.

For C/C++ client.

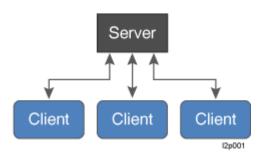
Remote host where Line2PdfServer is running: server1.

RMI port: 7201.

- 2. Enter the command line: java Line2PdfAgent -p 9500 -s server1-sp 7201.
- 3. To stop Line2PdfAgent enter: java Line2PdfAgent -p 9500 -stop.

# 3. Client/Server Model

- Client Native C/C++ Program
- Client and Server on the Same Machine
- Client and Server on Different Machines
- Enabling Debug Mode for Server and Agent



Client is C++ native code, while Server is Java code.

# Client supports:

- Windows
- Linux
- AIX
- Sun
- zOS/USS
- zLinux

Client/Server Model topology has 2 options:

- Client and Server on the same machine.
- Client and Server on different machines.

# Client Native C/C++ Program

# **Executable Program**

Windows: line2pdf.exe.

Other operating systems: 1 ine2pdf.

#### **Syntax**

# line2pdf [port number] <line2pdf arguments>

# Options:

#### port number (optional)

Socket port number. Default value: 8500.

## <line2pdf argument options>

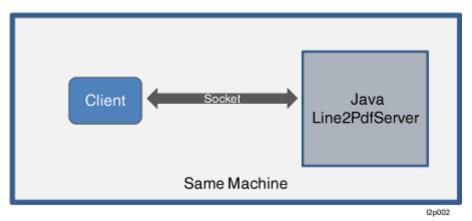
Same as Java command line.

#### Return code

O: if no error.

other numbers: if error occurs.

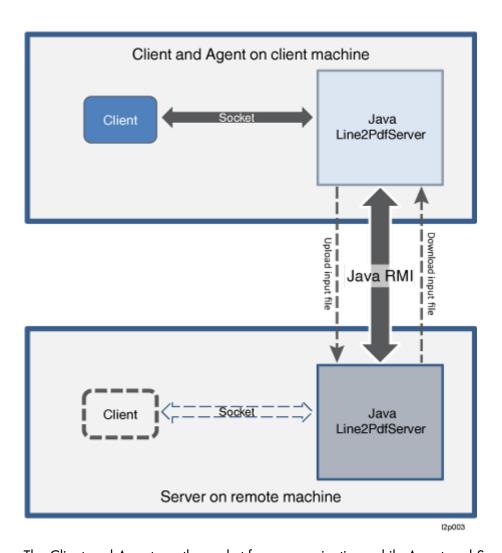
# Client and Server on the Same Machine



- 1. Client sends the command line arguments to server through socket communication.
- 2. Server sends a thread to transform line data to pdf and then sends the messages back to client to indicate if completed or failed.

# **Client and Server on Different Machines**

The Client/Server Model with Client and Server on different machines requires that the agent Line2PdfAgent is based on the client machine.



The Client and Agent use the socket for communication, while Agent and Server use Java RMI.

The Agent redirects the client requests to remote server. The LinePdfAgent receives the messages from remote server and then redirects the received messages to the client.

The Agent uploads the input line data to the server and downloads the output PDF file from the server, if transform is completed.

# **Enabling Debug Mode for Server and Agent**

To turn on the debug mode on Line2PdfServer and Line2PdfAgent use a hidden flag.

Flag —debug must be on the first argument. When debug mode is **On** debug information is printed on the console.

Example of Line2PdfServer on debug mode:

java Line2PdfServer **-debug** -start -p 7200 -r 7201