



Integrating with Other Applications

Version 3.11.2

Overview	1
Submitting jobs from other applications	2
Adding functions to workflows	3
Using Web Services	4
RICOH ProcessDirector symbol notation	5
File system mapping file for job tickets	6
Syntax for RICOH ProcessDirector control files	7
Database property names	8
Job properties that can be set from the job ticket	9

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

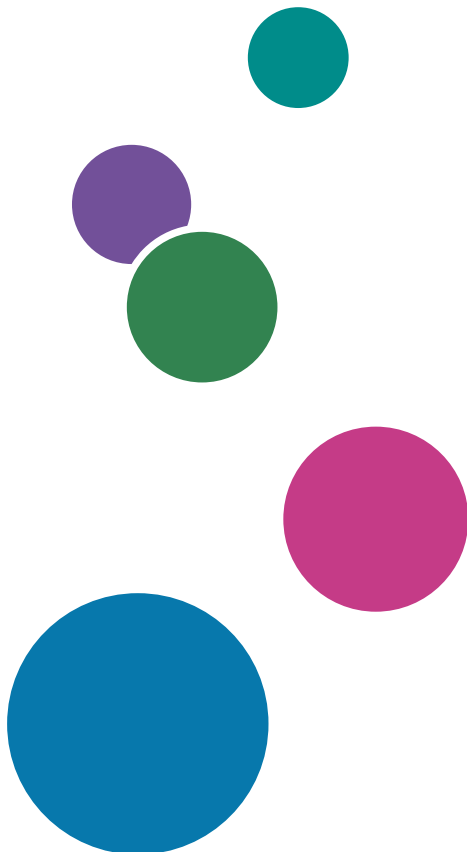


TABLE OF CONTENTS

Introduction

Important.....	4
Cautions regarding this guide.....	4
Publications for this product.....	4
How to read the documentation.....	5
Before using RICOH ProcessDirector.....	5
How to use the manuals.....	5
Related information.....	7
Symbols.....	7
Abbreviations.....	8
Trademarks.....	9

1 Overview

System objects.....	12
Secondary servers.....	13
Application server.....	14
Input devices.....	14
Step templates.....	15
Web services in RICOH ProcessDirector.....	17
Usage notes.....	17

2 Submitting jobs from other applications

Hot folder input devices.....	19
Batching methods.....	20
Files for batch jobs.....	32
Assigning workflows.....	40
Setting up hot folder input devices to process batch jobs.....	42
Configuring to use JDF job tickets.....	46
LPD input devices.....	50
Configuring to use the LPD protocol.....	53
Print commands used with the LPD protocol.....	59
Restricting use of the LPD protocol to submit jobs.....	61

3 Adding functions to workflows

Setting up external programs.....	69
External steps.....	72
External commands.....	72
Using RICOH ProcessDirector methods.....	73

Setting up step templates for external steps	85
Setting up step templates for external steps that use the command line or control files	85
Setting up step templates for external steps that use hot folders	86
Tuning step templates	88
Setting up workflows for external steps	89

4 Using Web Services

Using RICOH ProcessDirector REST API documentation	92
Preparing to submit jobs using web services	95

5 RICOH ProcessDirector symbol notation

6 File system mapping file for job tickets

7 Syntax for RICOH ProcessDirector control files

For rules	105
receive_jcl_jobtype.cfg	105
receive_lpd_jobtype.cfg	108
receive_lpd_pdf_jobtype.cfg	110
receive_text_jobtype.cfg	112
Using the -ooutbin parameter in JCL and LPD jobtype files	114
For Passthrough printers	115
For the Archive feature	116
Creating an associated properties file	116
Document properties file	122
Export results file	122

8 Database property names

For jobs	123
For workflows	147
For printers	149
For media	156
For input devices	159
For input files	166
For Notification	169
For servers	170
For step templates	171
For the system	174
For users	181

For groups	183
For locations	183
For security	184
9 Job properties that can be set from the job ticket	

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 improvements or changes 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.
- Throughout this publication, references to directory paths indicate the default paths only. If you install RICOH ProcessDirector or any of its components in a different location, including a different drive, you must adjust the paths accordingly.

For example, if you install RICOH ProcessDirector on the D: drive of a computer running a Windows operating system, replace C: with D: in the directory paths.

Publications for this product

The RICOH ProcessDirector publications CD includes the RICOH ProcessDirector publications.

Instruction manuals

These instruction manuals are included:

- *RICOH ProcessDirector: Integrating with Other Applications* (this publication)
This guide provides technical information about the ways that you can configure RICOH ProcessDirector to exchange data with other applications.
- *RICOH ProcessDirector for AIX, Linux, or Windows: Planning and Installing*
This guide explains planning and installation procedures for RICOH ProcessDirector on your operating system. The publications CD includes the version of this manual for your operating system: AIX, Linux, or Windows.
- *RICOH ProcessDirector: Installing Document Processing Features*

This guide explains how to install RICOH ProcessDirector features that control and track both jobs and the individual documents in jobs.

- *RICOH ProcessDirector: Using RICOH ProcessDirector Plug-in for Adobe Acrobat*

This guide explains how to use RICOH ProcessDirector Plug-in for Adobe Acrobat. You can use the Adobe Acrobat plug-in to define text, barcodes, images, and other enhancements in a PDF file. After you save your enhancements in a control file, RICOH ProcessDirector workflows can use the control file to make similar enhancements to PDF files.

- *Font Summary*

This guide explains font concepts and the different types of fonts in the InfoPrint Font Collection. The *Font Summary* is available only in English.

- *White Paper—Using the Enhance AFP Function*

This guide explains how to configure and use Enhance AFP control files. The guide is available only in English.

- The RICOH ProcessDirector readme file (`readme.html`)

This file tells you how to access the other publications. The readme file is available only in English.

- The RICOH ProcessDirector release notes

These release notes provide information about the RICOH ProcessDirector release, including new functions and updates; known limitations, problems, and workarounds; and code change requests. The release notes are available only in English.

You can also download English publications from the [RICOH Software Information Center](https://help.ricohsoftware.com/swinfocenter/) (<https://help.ricohsoftware.com/swinfocenter/>).

RICOH ProcessDirector Information Center

The Information Center contains topics that help administrators, supervisors, and operators learn about and use RICOH ProcessDirector. The Information Center is available from the user interface and provides quick navigation and search features.

Help

Field help is available on many screens to provide information for specific tasks and settings.

How to read the documentation

Before using RICOH ProcessDirector

This manual contains instructions and cautions for configuring RICOH ProcessDirector to exchange data with other applications. Before using RICOH ProcessDirector to exchange data with other applications, read this manual thoroughly and completely. Keep this manual handy for future reference.

How to use the manuals

Use the instruction manuals according to your needs.

To learn how to plan for, install, and start RICOH ProcessDirector:

See *RICOH ProcessDirector for AIX, Linux, or Windows: Planning and Installing*. The publications CD includes the version of this manual for your operating system: AIX, Linux, or Windows.

To learn about the functions and operations of RICOH ProcessDirector and its installed features:

See the RICOH ProcessDirector Information Center.

To learn how to set property values in the user interface:

See the field help.

To learn how to configure RICOH ProcessDirector to exchange data with other applications:

See *RICOH ProcessDirector: Integrating with Other Applications*.

To learn how to install a document processing feature:

See *RICOH ProcessDirector: Installing Document Processing Features*.

To learn how to use the functions and operations of RICOH ProcessDirector Plug-in for Adobe Acrobat

See *RICOH ProcessDirector: Using RICOH ProcessDirector Plug-in for Adobe Acrobat*.

Displaying the publications

The RICOH ProcessDirector publications are available on the publications CD, so you can access them before you install the application.

Note

- A PDF viewer, such as Adobe Acrobat Reader, is required to view the publications.


To access the RICOH ProcessDirector publications CD on Windows:

1. Insert the CD in the CD drive.
If the Windows system is configured to autorun CDs, Windows Explorer opens automatically to show the contents of the CD.
2. If Windows Explorer does not start automatically, open it and display the contents of the CD drive.
3. Open the `readme.html` file for information about the contents of the CD.

Some of these publications are also available from RICOH ProcessDirector user interface.

Note

- You must log in to the RICOH ProcessDirector user interface to view the publications.


On the top menu bar of the RICOH ProcessDirector user interface, click the  button and select one of the following publications to download:

- *RICOH ProcessDirector: Integrating with Other Applications*
- *RICOH ProcessDirector: Installing Document Processing Features*
- *RICOH ProcessDirector: Using RICOH ProcessDirector Plug-in for Adobe Acrobat*
- *RICOH ProcessDirector: Release Notes*

Displaying the Information Center

The RICOH ProcessDirector Information Center is available from the user interface.

To display the Information Center:

- On the top menu bar of the RICOH ProcessDirector user interface, click the  button and select **Help**.
- If you are not logged in to RICOH ProcessDirector, enter this URL in the address bar of your browser:
`http://hostname:15080/pdhelp/index.jsp`
In the URL, *hostname* is the host name or IP address of the computer where RICOH ProcessDirector is installed.

In addition, you can bookmark the location of the Information Center in your browser and open it at any time outside of RICOH ProcessDirector.

Information about use the functions and operations of features are available only when the features are installed in the system.

Related information

For information about our products, see:

- [RICOH website](https://ricohsoftware.com) (https://ricohsoftware.com)
- [RICOH Software Information Center](https://help.ricohsoftware.com/swinfocenter/) (https://help.ricohsoftware.com/swinfocenter/)

For information about related products, see:

- *InfoPrint Manager: PSF and Server Messages*, G550-1053
- *RICOH InfoPrint XT for Linux: Installation and User's Guide*, G550-20375
- *RICOH InfoPrint XT for Windows: Installation and User's Guide*, GLD0-0025
- *AFP Conversion and Indexing Facility User's Guide*, G550-1342
- *Print Services Facility for z/OS: AFP Download Plus*, S550-0433
- *Print Services Facility for z/OS: Download for z/OS*, S550-0429

Symbols

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

Important

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

Note

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

Bold

Bold type indicates the names of dialogs, menus, menu items, settings, field labels, buttons, and keys.

Italic

Italic type indicates the titles of manuals and variables that you must replace with your own information.

Monospace

Monospace type indicates computer input and output.

Abbreviations

AFP

Advanced Function Presentation

API

Application Programming Interface

CSV

Comma-Separated Values

DNS

Domain Name System

HTTP

Hyper Text Transfer Protocol

IP

Internet Protocol

JDF

Job Definition Format

LPD

Line printer daemon

PDF

Portable Document Format

PSF

Print Services Facility

REST

Representational State Transfer

SMIT

System Management Interface Tool

SOAP

Simple Object Access Protocol

SSL

Secure Sockets Layer

YaST

Yet Another Setup Tool

Trademarks

RICOH ProcessDirector™ and RICOH InfoPrint Manager™ are trademarks of Ricoh Company, Ltd. in the United States, other countries, or both.

Adobe, Reader, and PostScript are either registered trademarks or trademarks of Adobe Systems, Inc in the United States and/or other countries.

IBM, AIX, MVS, Print Services Facility, and z/OS are either registered trademarks or trademarks of International Business Machines Corporation in the United States and/or other countries.

Impostrip® is a registered trademark of Ultimate TechnoGraphics Inc.

Kodak is a registered trademark of Eastman Kodak Company.

Linux is a registered trademark of Linus Torvalds.

MarcomCentral is a registered trademark of MarcomCentral, a Ricoh Company.

Microsoft, Windows, Windows Server, and Internet Explorer are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group.

Xerox is a registered trademark of Xerox Corporation.

Thales Group code: The license management portion of this Licensee Application is based on one or more of the following copyrights:

Sentinel RMS
Copyright 1989-2022 Thales Group
All rights reserved.

Sentinel Caffé (TM)
Copyright 2008-2022 Thales Group
All rights reserved.

Sentinel EMS
Copyright 2008-2022 Thales Group
All rights reserved.

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

- Windows 10:

Microsoft Windows 10 Pro

Microsoft Windows 10 Enterprise

- The product name of Windows 11 is:

Microsoft Windows 11 Pro

- Windows Server 2012 R2:

Microsoft Windows Server 2012 R2 Standard

Microsoft Windows Server 2012 R2 Enterprise

- Windows Server 2016:

Microsoft Windows Server 2016 Standard

Microsoft Windows Server 2016 Datacenter

- Windows Server 2019:

Microsoft Windows Server 2019 Standard

- The product name of Windows Server 2022 is:

Microsoft Windows Server 2022 Standard

Other product names used herein are for identification purposes only and might be trademarks of their respective companies. We disclaim any and all rights to those marks.

1. Overview

- System objects
- Web services in RICOH ProcessDirector

RICOH ProcessDirector provides several mechanisms for interacting with other applications. The mechanism you use depends on your goal and on the capabilities of the application.

The general goals that you might want to accomplish when you use RICOH ProcessDirector with another application are:

1. Submit a job for processing and printing.

Your application can copy print files to hot folder input devices, which are directories that RICOH ProcessDirector monitors at configurable intervals. RICOH ProcessDirector accepts the print files and converts them into print jobs.

You can send jobs to hot folders using a variety of methods. If you submit Job Definition Format (JDF) job tickets with your print files, the hot folder input device can use the job ticket to set job properties. The JDF job ticket stays with the job during processing. Most job properties and the values in the job ticket are synchronized whenever the job ticket is requested. You can also send jobs into a hot folder with the JDF batching method with the PDF, JDF and JMF information in a single MIME package. The JMF must reference the name of the hot folder as the DeviceID.

If your application uses the line print (LPR) function to transmit files, it can send them to a line printer daemon (LPD) input device. LPD input devices are similar to hot folder input devices, but they can only receive files that are sent using LPR.

2. Add processing capabilities that are not available in RICOH ProcessDirector to your print workflows.

If your application provides processing that can be incorporated into print workflows (such as address cleansing, data stream transformation, or imposition), you can create steps that send the print job from RICOH ProcessDirector to your application, wait for a valid return code, and then continue processing in RICOH ProcessDirector.

If your application monitors hot folders, the steps you create can copy print jobs to and from those hot folders. If your application has a command line, the step can run the appropriate command and wait for the resulting file to be returned.

If you submit jobs with JDF job tickets, the application can use values in the job ticket, even if they are not supported by RICOH ProcessDirector. If the application updates values in the job ticket, the new values can be reflected in RICOH ProcessDirector.

RICOH ProcessDirector provides features that make it easier to integrate with other programs into workflows.

RICOH ProcessDirector Feature	Associated Program
Avanti Slingshot Connect	Avanti Slingshot
Quadient Inspire Connect	Quadient Inspire Designer
Marcom Central Connect	Marcom Central
PitStop Connect	Enfocus PitStop Server 10 or higher
Ultimate Impostrip® Connect	Ultimate Impostrip®

RICOH ProcessDirector also provides the Postal Enablement feature. The Postal Enablement feature lets you connect to your choice of external postal software. Examples of postal software include TEC Mailing Solutions MailPreparer or the Bell and Howell BCC Mail Manager.

3. Access RICOH ProcessDirector using a REST API.

You can use the REST APIs provided with RICOH ProcessDirector to integrate with the Web services for your application. With these REST APIs you can make requests and issue commands using Web services.

Making requests using Web services includes retrieving the status of objects in the system, property values, and log files, among other things. For example, if you want to collect the same information about various jobs in the system for use in a report or to display in a different user interface, you can create a Web services request to ask for that information.

Issuing commands using Web services includes connecting and disconnecting input devices, enabling and disabling objects, and setting object properties. For example, if you want to enable and connect an input device, you can create a Web service request that issues the correct commands.

Later chapters provide more information about each goal and the ways to achieve them. Before you read those chapters, make sure that you understand the basic information about RICOH ProcessDirector that this chapter explains.

System objects

The RICOH ProcessDirector system contains objects that represent hardware, software, files, paper, and processes. You can configure the properties of the objects to meet your needs.

The base product provides many kinds of system objects, such as:

- **Primary server**
The system that RICOH ProcessDirector is installed on. This system holds most of the product components, including the internal database, Web server, print driver, help system, and message logs.
- **Secondary servers**
Other AIX or Linux systems that can connect to the primary server. Secondary servers can be used to offload resource-intensive process steps and improve processing speed.

Secondary servers can also be used to access applications that run on different platforms. For example, a print vendor runs a primary server on an AIX system. They want to use an impositioning application in their workflow, but it only runs on Linux. They can install a RICOH ProcessDirector secondary server on a Linux system and configure their workflows to send print jobs through that application.

Secondary servers can only be used with RICOH ProcessDirector for AIX and RICOH ProcessDirector for Linux.
- **Application servers**
Application servers share the database that the primary server uses and work with the primary server to process jobs effectively. Application servers are installed on Windows computers. They can run steps that require applications on Windows and can be used as parent servers for input devices, Ricoh PDF printers, Custom PDF printers, Ricoh TotalFlow printers, Kodak printers, Xerox printers, and Passthrough printers. They cannot run other steps (such as **PrintJobs**) or be used as the parent server for other kinds of printers.

Application servers can only be used with RICOH ProcessDirector for AIX and RICOH ProcessDirector for Linux.

- Step templates

Step templates are the basic building blocks for processing in the RICOH ProcessDirector system. Each step template contains code that does a specific action, such as setting job properties, transforming from one data stream to another, or retaining a job for a given time period.

A step template becomes instantiated as a step when it is added to a workflow.

- Workflows

A workflow is a set of steps, arranged and configured to meet the needs of the print provider. A print job moves from step to step in a workflow until it completes the workflow without error or reaches the end of a processing path. In some cases, a job might be processed through more than one workflow. Workflows can be linear or conditional. In a linear workflow, each step can receive jobs from one step and send jobs to one step. In a conditional workflow, each step can receive jobs from one or more steps and send jobs to one or more steps. When a step sends jobs to multiple steps, the connectors between the steps specify conditions that control which step each job is sent to.

★ **Important**

- When you use the **Workflow Editor** to edit a job type created in a previous version of RICOH ProcessDirector, the job type becomes a workflow. It can no longer be opened on the **Administration** page by clicking **Workflow** → **Job Types**.

- Input devices

Input devices receive input files, create print jobs from those files, assign the print jobs to workflows, and submit the jobs for processing. There are several kinds of input devices. You choose which kind to use based on how jobs are submitted to them.

- Printers

Printers represent the printer hardware that is connected to the system and that can receive jobs from RICOH ProcessDirector.

Optional features and extended features can add other types of objects to the system.

When RICOH ProcessDirector interacts with other applications, the key objects are secondary servers, application servers, input devices, and step templates.

For a list of available system objects, you can use **objecttypes** API found in the utility section of the RICOH ProcessDirector REST API documentation. See [Using RICOH ProcessDirector REST API documentation, p. 92](#) for more information.

Secondary servers

Secondary servers let you distribute processing, so that your RICOH ProcessDirector system runs more efficiently.

You can use secondary servers to do some of the more processing-intensive steps in your various workflows and to prevent the primary server from becoming overloaded. The primary server works with the secondary servers to coordinate the movement of all jobs across the system. The secondary servers do not have their own embedded databases for storing system information. All the servers work with the database that is installed on the primary computer.

Secondary servers can manage all types of RICOH ProcessDirector objects, such as input devices, steps, and printers. They can also run external programs that RICOH ProcessDirector accesses through external steps. External programs can do more processing or special types of processing.

For installations that have many printers and a high volume of jobs, adding secondary servers to the system can increase job throughput. The throughput increases because each server has a smaller number of printers and jobs to monitor and control.

Application server

If your workflows require processing by applications that run on Windows, you can access those applications by installing an application server on a Windows computer.

Application servers work with primary computers running AIX or Linux. They do not work with primary computers running Windows.

Application servers share the database that the primary server uses and work with the primary server to process jobs effectively. Because application servers are installed on Windows computers, they can run steps that require applications on Windows. They cannot run other steps, such as **PrintJobs**.

Application servers can be the parent servers for Ricoh PDF printers, Custom PDF printers, and Passthrough printers. They cannot be the parent servers for other kinds of printers.

You can install application servers on these operating systems:

- Windows 10 Pro or Enterprise 64-bit
- Windows 11 Pro
- Windows Server 2019 64-bit
- Windows Server 2022 64-bit

Input devices

Input devices receive input files, create jobs, and assign the jobs to workflows for processing.

RICOH ProcessDirector provides these types of input devices:

- Hot folder input devices are directories that the system monitors. When a print job is copied or sent to a Hot folder, the system notices it and begins to process it.
- Line printer daemon (LPD) input devices receive jobs that are submitted using the LPD protocol. LPD input devices specify a control file, `receive_lpd_jobtype.cfg` or `receive_lpd_pdf_jobtype.cfg`, to convert LPD control file parameters for the job to a text-based job property file.
- REST web service input devices call a Representational State Transfer web service to retrieve XML or JSON files from a third-party application. The input device can create a job containing the entire XML or JSON file, or it can examine the file using an XPath or JSONPath expression. If the input device finds XML elements or JSON objects specified by the expression, it can create a single job containing the entire file. Alternatively, it can create one or more jobs, each containing a portion of the file. This device type is only available if the Web Services Enablement feature is installed.

- SOAP web service input devices call a Simple Object Access Protocol web service to retrieve XML files from a third-party application. The input device can create a job containing the entire XML file, or it can examine the file using an XPath expression. If the input device finds XML elements specified by the expression, it can create a single job containing the entire file. Alternatively, it can create one or more jobs, each containing a portion of the file. This device type is only available if the Web Services Enablement feature is installed.

The AFP Support feature provides Download input devices. These input devices receive data sets from Download for z/OS or AFP Download Plus. For information about submitting jobs to Download input devices, see *Print Services Facility for z/OS: Download for z/OS (S550-0429)* or *Print Services Facility for z/OS: AFP Download Plus (S550-0433)*.

Hot folder input devices

Hot folder input devices are directories on a primary or secondary server that the system monitors for input files.

To move input files into hot folders, you can use a copy command, file transfer protocol (FTP), secure copy protocol (SCP), or any other method that lets you transfer files to a specified location.

RICOH ProcessDirector provides several predefined hot folder input devices that you can copy and modify to fit your environment.

LPD input devices

LPD input devices receive jobs that are submitted using an LPR command. LPD input devices specify a control file, `receive_lpd_jobtype.cfg` or `receive_lpd_pdf_jobtype.cfg`, to convert LPD control file parameters for the job to a text-based job property file.

RICOH ProcessDirector provides several predefined LPD input devices that you can use.

Download input devices

Download input devices receive jobs from Download for z/OS® or AFP Download Plus. Download input devices specify a control file to change JCL parameters for a job to a job properties file in RICOH ProcessDirector property name=value format that the workflow uses.

The AFP Support feature provides several predefined download input devices that you can use.

Step templates

Step templates are separate units of processing code. They can be assembled into workflows based on what you need to do to an input file.

RICOH ProcessDirector provides many step templates that you can incorporate into workflows; optional features can add even more. You can either use those step templates as they are, or copy and modify them to work in your environment.

 Note

Some step templates cannot be copied; but you can edit their properties.

You cannot write your own step templates and include them in workflows. However, if you write a script or other program that can be invoked from the command line, you can access it using the `RunExternalProgram` step template.

Some step templates are particularly important for interacting with other applications:

- **SetJobType** step templates

Input devices use `SetJobType` steps to determine which workflow to use to process a print job. The `SetJobType` step templates are:

- **SetJobTypeFromFileName**

Uses a pattern-matching string to set the workflow from a portion of the input-file name, such as the file extension. For example, all jobs with the file extension **.AFP** can be sent to the workflow **AFP** or all jobs with the extension **.pdf** can be sent to the workflow **pdf**. The step can also convert an optional overrides file submitted with a job to a text-based file, *jobID.overrides.text*, which the **SetJobPropsFromTextFile** step uses to set job properties.

- **SetJobTypeFromRules**

Uses a control file to select the workflow based on the value of a parameter in an overrides files that accompanies the input file. The step can also convert an optional overrides file submitted with a job to a text-based file, which the **SetJobPropsFromTextFile** step uses to set job properties.

Print system administrators might want to coordinate the output of a print submission application with the values in these steps.

- **SubmitInputFiles** step template

Input devices use the **SubmitInputFiles** step to send print jobs to the first step in their workflows. It submits a group of input files as a parent job and child jobs according to the batching method specified. It submits a single input file as a single job of the child workflow, without a parent job.

- **SetJobPropsFromTextFile** step template

The **SetJobPropsFromTextFile** step is usually the first step in a workflow. It uses a text file that accompanies the input file to set job properties. The text file specifies which properties to set. Values in the text file overwrite any values that have been set previously. After processing an optional *jobID.overrides.text* file, the step looks for an optional Job Definition Format (JDF) job ticket file, *jobID.overrides.jdf*, and uses it to set additional properties of the job.

- **External steps**

External steps send print jobs to applications outside of RICOH ProcessDirector for processing such as data stream transforms, address cleansing, and impositioning. The external step templates are:

- **RunExternalProgram**

Submits the print job to an external program using a command line interface.

- **RunHotFolderApplication**

Puts the print job in a hot folder that another application monitors. That application picks up the print job, processes it according to its configuration settings, and puts it into an output folder. RICOH ProcessDirector monitors that output folder and retrieves the job when it is ready.

Web services in RICOH ProcessDirector

RICOH ProcessDirector provides Web services that use Representational State Transfer (REST) software architecture principles so you can access its functions from other applications.

RESTful Web services send requests and receive results using the Hypertext Transfer Protocol (HTTP). Each Web service has a Universal Resource Indicator (URI) associated with it. When you create requests, you modify the URI to include specific information about the object that you are interested in or the action that you want to do. These modifications can be simple, such as inserting the name of a printer object into the URI, or highly complex, such as using Extensible Markup Language (XML) to describe the criteria to use when searching for objects and how they should be sorted when they are returned. If you code XML elements to include in your requests, they must be Universal Resource Locator (URL)-encoded before they are added to the URI.

With RICOH ProcessDirector, you can use Web services to request information about print jobs and other objects in the system. You can also use them to set properties and do actions such as enabling and disabling objects, modifying properties, and starting and stopping printers.

Usage notes

The Web services provided with RICOH ProcessDirector follow specific conventions. When you integrate Web services with your application, refer to these usage notes to make sure that your implementation functions properly.

- You must use the **GET /users/login/** Web service to establish a connection and receive a credential token from the RICOH ProcessDirector server before you can use any other Web services. All of the other Web services require that you include a credential token in the request header of the URI. If you do not, you receive an immediate 401 Unauthorized HTTP error.
- You cannot translate actions and URIs into other languages because they are fixed-character strings. For example, "**token:**" remains the same for all languages.
- Record sets retrieved might change between invocations of any service, especially when you use sorting or filtering functions; successive identical requests might not yield the same records.
- If you activate Secure Sockets Layer (SSL) or Transport Layer Security (TLS) support, RICOH ProcessDirector uses the SSL protocol for all HTTP communications, including Web services. If SSL or TLS support is active, you must use a Web services client that supports SSL or TLS to invoke Web services.
- RICOH ProcessDirector Web services only return JavaScript Object Notation (JSON) data. XML is not supported.

2. Submitting jobs from other applications

- Hot folder input devices
- LPD input devices

If your application is a document composition or a job submission tool, you can use hot folder or LPD input devices to get print jobs into RICOH ProcessDirector to be processed and printed.

If your application can use REST web services, you can use the **submitFile** utility to submit a file to a workflow or hot folder. See [Preparing to submit jobs using web services, p. 95](#) for more detailed information about this implementation.

The key functions of input devices are to:

- Receive jobs into the system.
- Assign jobs to workflows.
- Send jobs to the first step in the workflow to begin processing.

Hot folder input devices can receive a variety of print jobs, including those that are submitted with JDF job tickets. LPD input devices can receive print jobs submitted using a line printer (LPR) command.

Note

Neither type of input device returns information to the submitting application when print jobs are received. For example, if you send a print job to a hot folder input device, the hot folder does not send the job number back to the submitting application so it can monitor the progress of the job.

Hot folder input devices

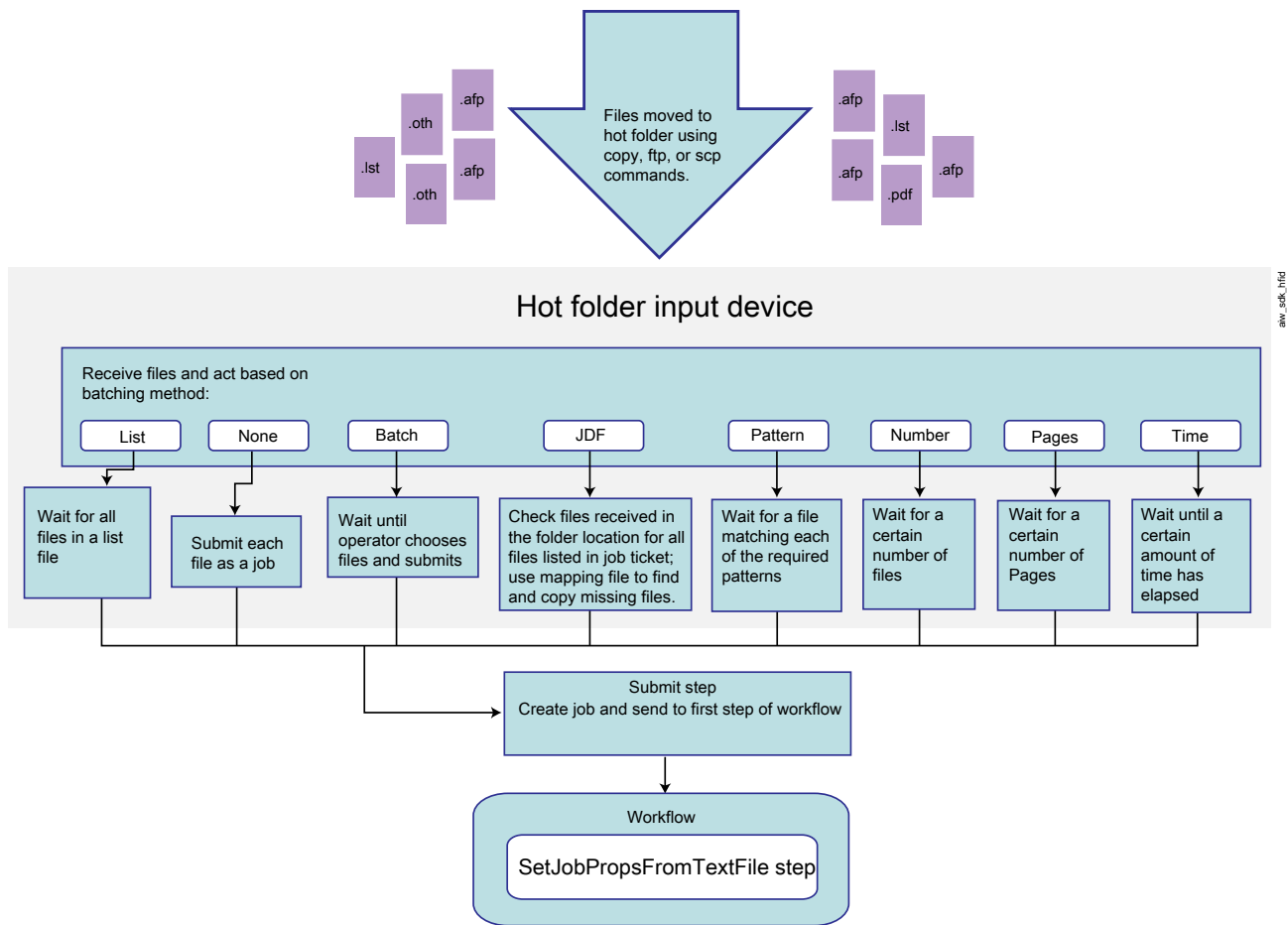
Hot folder input devices can receive print jobs that include a single file, jobs that include multiple files, and jobs that include JDF job tickets.

When you configure a hot folder, you determine what types of jobs it should handle. Then, you have to make sure that jobs are submitted to the correct hot folder so they are processed correctly. Otherwise, print jobs can encounter errors before they even start to process through a workflow.

Input files can be processed individually or as groups. To process them as groups, you can create a batch. Some batching methods require additional supporting files that must be submitted with the input files.

The figure [Hot folder input device processing, p. 20](#) shows how a hot folder input device processes jobs. The input device acts on jobs based on the value of the batching method and passes the job to the Submit step.

Hot folder input device processing



Permissions

All users who submit jobs to a hot folder (including the user IDs that job submission applications run under) must have the appropriate permissions for that folder. The easiest way to make sure that users have the correct permissions is to add all submitters to the RICOH ProcessDirector group (**aiwgrp1** is the default), which is created when RICOH ProcessDirector is installed.

Batching methods

The way that hot folder and SFTP input devices submit jobs is determined by the **Batching method** property of the input device.

By default, using any batching method except **None** causes an input device to create jobs as groups that use the parent/child structure. A parent job contains no data; it is a container that maintains the relationship between other jobs. Those jobs are child jobs. Each input file that a batching method includes in a group becomes a child job.

For all batching methods except **None**, **JDF**, and **List**, RICOH ProcessDirector for Windows processes child jobs in the order that the files were last modified. The Windows **Date Modified** value is shown as the value of the **Time submitted** job property. Because the **Date Modified** value does not change

when the file is received by the input device, a file created in 2010 and submitted in 2011 shows a **Time submitted** from 2010.

Note

- If you set the **Create .zip file** property of an input device to **Yes**, the input device does not create groups of jobs that use the parent/child structure. Instead, the input device gathers all the input files in the group and creates a ZIP file to hold them. The ZIP file is submitted as a single job. You must have a step in the workflow that unzips the file unless the other steps can process a file in ZIP file format.
- You can submit all the input files in an input device with the **Batch all** action on the input device as long as the **Batching method** is not **JDF** or **Pattern**. The **Batch all** action does not wait for the **Polling interval** to be reached before creating the jobs.

Batch

When the batching method is **Batch**, the input device submits one or more files as a group, based on the files that an operator selects.

The **Batch** batching method prevents the input device from submitting jobs immediately. The input device receives files but does nothing until an operator selects one or more data files and clicks **Make batch**. The input device creates a list file that contains the names of all the data files and submits the job.

When you use the **Batch** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the data files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs.
- **Data patterns:** A pattern that matches some part of the file name of your files. For hot folder and SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.

JDF

When the batching method is **JDF**, the hot folder or SFTP input device submits one or more files as a group, based on the contents of the job ticket.

The hot folder or SFTP input device looks for a job ticket containing a list of all the files that make up the job. When the hot folder or SFTP input device finds a job ticket, it reads the ticket and looks for all the files; as soon as it finds all the files, it submits them as a single job. The files print in the order specified in the job ticket.

The file name of the job ticket must match the pattern specified in the **JDF patterns** property. The default pattern matches files that have the extension `.jdf`.

The list of files in the job ticket might look like this:

```
myfile1.pdf
myfile2.pdf
myfile3.pdf
another.pdf
```

The job ticket might also contain information that is used to set job properties.

When you use the **JDF** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs. If the job ticket specifies different settings for different input files and the **Create .zip file** property is set to **Yes**, the settings for the first input file name in the job ticket are used for all input files.
- **JDF patterns:** A pattern that matches some part of the file name of your job tickets. For all input devices except HotFolderJDF the default value is null. For HotFolderJDF this value is the regular expression `.*\jdf$`.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.

List

When the batching method is **List**, the hot folder or SFTP input device submits one or more files as a group, based on the contents of a list file that is received by the input device.

The hot folder or SFTP input device looks for a text-based **list file** containing a list of all the files that make up the job. When the hot folder or SFTP input device finds a list file, it reads the list and looks for all the files; as soon as it finds all the files, it submits them as a single job. The files print in the order specified in the list file.

The list file must meet these criteria:

- The file name must match the pattern specified in the **List patterns** property. The default pattern matches files that have the extension `.lst`.
- Each file in the print job must be on a separate line in the list file.
- The file names listed in the list file must include only file names, not path information; all files must be in the same directory.

The contents of the list file might look like this:

```
afpinput1.afp
afpinput2.afp
afpinput3.afp
inputfile.afp
```

When you use the **List** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint

- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs.
- **List patterns:** A pattern that matches some part of the file name of your list files.
By default, this value is the regular expression `.*1st$`.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.
If a print file is listed in the list file, but the file name does not match the **Data patterns** value, the input device does not identify the file as a print file. Because a listed print file appears to be missing, the input device waits for it and does not create the job.
- **Overrides patterns:** if you use overrides files, a pattern that matches some part of the file name of your overrides files.

None

When the batching method is **None**, the hot folder or SFTP input device submits every file as a separate print job.

Number

When the batching method is **Number**, the hot folder or SFTP input device submits a specific number of files as a group. Only files that match the **Data patterns** property of the hot folder or SFTP input device are counted and submitted.

The hot folder or SFTP input device parses the file name and determines if it matches the value of the **Data patterns** property. If the name matches the pattern, the hot folder or SFTP input device counts the file. When the number of files reaches the value set for the **Number of files to batch** property, the hot folder or SFTP input device submits the files as a single group.

Note

- RICOH ProcessDirector for Windows processes child jobs in the order that the files were last modified. The Windows **Date Modified** value is shown as the value of the **Time submitted** job property. Because the **Date Modified** value does not change when the file is placed in the input device, a file created in 2010 and submitted in 2011 shows a **Time submitted** from 2010.

When you use the **Number** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs.

- **Number of files to batch:** The number of files that the hot folder or SFTP input device waits to receive before submitting the files as a single group.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.

Number of sets

2

When the batching method is **Number of sets**, the hot folder or SFTP input device submits one or more jobs after a specific number of sets of files are received. Only complete sets of files are counted and submitted.

For a set to be complete, all of these conditions must be true:

- A file matching the value in the **Data patterns** property is present.
- If a value is entered for the **Overrides patterns** or **JDF patterns** properties, a file matching one or both of those values is present.
- If there are any entries specified for **File patterns**, a file matching each value is present.

When the number of complete sets reaches the value set for the **Number of files to batch** property, the hot folder or SFTP input device submits the sets as a group with one set in each child job. If the **Create .zip file** property is set to **Yes**, a single job containing all of the files in all of the sets is submitted in ZIP file format.

When you use the **Number of sets** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all of the files are combined as a ZIP file and submitted as a single job. If this value is set to **No**, each set is submitted as a child job.
- **Number of files to batch:** The number of sets that the hot folder or SFTP input device waits to receive before submitting them as a single group.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.
- **File Patterns:** A collection of patterns that can define files that are required to be included in a set.
- **JDF patterns:** One or more pattern-matching strings used to identify Job Definition Format (JDF) job tickets. If this property has a value specified, a JDF job ticket is required for the set to be complete.
- **Overrides patterns:** One or more pattern-matching strings used to identify files that contain values for job properties.

Pages

When the batching method is **Pages**, the hot folder or SFTP input device submits one or more PDF files as a group, based on the number of pages in the files that are received by the input device. This method is only valid for PDF files.

The hot folder or SFTP input device examines PDF files as they arrive and determines how many pages they contain. The hot folder or SFTP input device maintains a count of the total number of PDF pages currently in the input device. When the total number of PDF pages matches or exceeds the value set for the **Number of pages to batch** property, the hot folder or SFTP input device submits a group of files.

Note

- RICOH ProcessDirector for Windows processes child jobs in the order that the files were last modified. The Windows **Date Modified** value is shown as the value of the **Time submitted** job property. Because the **Date Modified** value does not change when the file is placed in the input device, a file created in 2010 and submitted in 2011 shows a **Time submitted** from 2010.

The PDF file that makes the total number of pages exceed the value set for the **Number of pages to batch** property is only included in the group if the **Exceed pages to batch** property is set to **Yes**. If the **Exceed pages to batch** property is set to **No**, this PDF file remains in the hot folder or SFTP input device as the first set of pages for the next batch.

If a single file contains more pages than the value set for the **Number of pages to batch** property and the **Exceed pages to batch** property is set to **Yes**, the file is submitted for printing, either as a batch with the rest of the PDF files that are waiting to print or as a batch that contains only one file. However, if the **Exceed pages to batch** property is set to **No**, the file cannot be submitted. Processing for the hot folder or SFTP input device stops until that input file is deleted, the value of the **Number of pages to batch** property is increased to at least the number of pages in the file, or the **Exceed pages to batch** property is changed to **Yes** so the file can be submitted.

Note

- The **Pages** batching method does not support encrypted or password-protected PDF files. If an encrypted or password-protected PDF file is submitted to a hot folder or SFTP input device that uses the **Pages** batching method, RICOH ProcessDirector issues an error message because it cannot open the file to count the number of pages.

When you use the **Pages** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs.
- **Number of pages to batch:** the number of pages that the hot folder or SFTP input device waits to receive before submitting the files as a single group.
- **Exceed pages to batch:** If this property is set to **Yes**, a PDF file that contains enough pages to make the total number of pages in the hot folder or SFTP input device exceed the value set for **Number of pages to batch** property will be submitted with the rest of the files. If this property is set

to **No**, this file will remain in the hot folder or SFTP input device as the first set of pages for the next batch.

- **Data patterns:** a pattern that matches PDF file names, such as `.*pdf$` or `.*PDF$`.

Pages in sets

2

When the batching method is **Pages in sets**, the hot folder or SFTP input device submits one or more jobs after a set of PDF files with a specified page count is received by the input device. This method is only valid for PDF files.

For a set to be complete, all of these conditions must be true:

- A file matching the value in the **Data patterns** property is present.
- If a value is entered for the **Overrides patterns** or **JDF patterns** properties, a file matching one or both of those values is present.
- If there are any entries specified for **File patterns**, a file matching each value is present.

The hot folder or SFTP input device examines the PDF files as they arrive and determines how many pages they contain. The hot folder or SFTP input device maintains a count of the total number of PDF pages in complete sets currently in the input device. When the total number of PDF pages in complete sets matches or exceeds the value set for the **Number of pages to batch** property, the hot folder or SFTP input device submits the sets as a group with one set in each child job. If the **Create .zip file** property is set to **Yes**, a single job containing all of the files in all of the sets is submitted in ZIP file format.

The set containing a PDF file that makes the total number of pages exceed the value set for the **Number of pages to batch** property is only included in the group if the **Exceed pages to batch** property is set to **Yes**. If the **Exceed pages to batch** property is set to **No**, this set remains in the hot folder or SFTP input device as the first set for the next batch.

If a single set contains more pages than the value set for the **Number of pages to batch** property and the **Exceed pages to batch** property is set to **Yes**, the set is submitted for printing, either as a batch with the rest of the sets that are waiting to print or as a batch that contains only one data file. However, if the **Exceed pages to batch** property is set to **No**, the set cannot be submitted. Processing for the hot folder or SFTP input device stops until that input file is deleted, the value of the **Number of pages to batch** property is increased to at least the number of pages in the set, or the **Exceed pages to batch** property is changed to **Yes** so the set can be submitted.

↓ Note

- The **Pages in sets** batching method does not support sets with encrypted or password-protected PDF files. If a set with an encrypted or password-protected PDF file is submitted to a hot folder or SFTP input device that uses the **Pages in sets** batching method, RICOH ProcessDirector issues an error message because it cannot open the file to count the number of pages.

When you use the **Pages in sets** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles

- **Create .zip file:** If this property is set to **Yes**, all of the files are combined as a ZIP file and submitted as a single job. If this value is set to **No**, each set is submitted as a child job.
- **Number of pages to batch:** The number of pages that the hot folder or SFTP input device waits to receive before submitting the files as a single group.
- **Exceed pages to batch:** If this property is set to **Yes**, a PDF file that contains enough pages to make the total number of pages in the hot folder or SFTP input device exceed the value set for **Number of pages to batch** property will be submitted with the rest of the files. If this property is set to **No**, this file will remain in the hot folder or SFTP input device as the first set of pages for the next batch.
- **Data patterns:** A pattern that matches PDF file names, such as `.*pdf$` or `.*PDF$`.
- **File Patterns:** A collection of patterns that can define files that are required to be included in a set.
- **JDF patterns:** One or more pattern-matching strings used to identify Job Definition Format (JDF) job tickets. If this property has a value specified, a JDF job ticket is required for the set to be complete.
- **Overrides patterns:** One or more pattern-matching strings used to identify files that contain values for job properties.

Pattern

When the batching method is **Pattern**, the hot folder or SFTP input device copies one print file and its related files to the spool file directory for the job as soon as it has all the required files. All these files must match the values set for the **Data patterns** property and the properties on the **Batching** tab of the input device properties notebook.

When you use the **Pattern** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** No

For example, you might want to send a JDF job ticket file along with an overrides file and a data file for a job. If a job ticket input file ends in `.jdf` and should have spool file usage "overrides" and spool file type "jdf", you can associate those three patterns with each other. Using RICOH ProcessDirector regular expression syntax, you define a series of input file name patterns that a hot folder or SFTP input device will recognize and include in the new job it creates. This table shows a simple example of files that can be processed together using pattern-matching.

Data pattern	File pattern	File usage	File type	Spool file type	Example matching file name	Example spool file name	Notes
.*\$.*jdf\$	overrides	Other	jdf	ab-c000317.jdf	/aiw/aiw1/spool/default/10000143/10000143.overrides.jdf	JDF job ticket
.*\$.*oth\$	overrides	Other	txt	ab-c00317.oth	/aiw/aiw1/spool/default/10000143/10000143.overrides.txt	Overrides file
.*\$.*pdf\$	print	Data	pdf	ab-c00317.pdf	/aiw/aiw1/spool/default/10000143/1000143.print.pdf	PDF file

You can use the **Data patterns** property and the **File Patterns** property to create precise file name and file type matches. If you want to use parts of the pattern from the **Data patterns** field as part of a file pattern, surround those parts of the **Data patterns** expression in parentheses, forming a backreference that you express in the **File Patterns** field with a backslash and a number. This table shows the results of pattern matching using a backreference:

Tab	Field	Value	Description
Batching	Data patterns	(.*)\.pdf	The expression (.*) defines a backreference to the data file name without the extension. When a data file name matches this pattern, all the characters that match (.*) are assigned to backreference \1.
Batching	File pattern	\1\.jdf	Matches the data file name, but changes the extension to .jdf. Job1.pdf and Job1.jdf are considered a match and are included in the job, but Job2.jdf does not match.

In a more complex example:

Tab	Field	Value	Description
Batching	Data patterns	(abc)(def)\.pdf,.*-(12)-.*\ .pdf	<p>The data file must match one of these two comma-separated patterns.</p> <p>Data pattern 1: \1 = abc and \2 = def</p> <p>The expressions (abc) and (def) define backreferences to the data file name. The extension must be .pdf.</p> <p>A data file that matches pattern 1 is abcdef.pdf.</p> <p>Data pattern 2: \1 = 12</p> <p>The expression (12) defines a backreference requiring that the data file is for the month of December (assuming the date format of a data file is <i>year-month-day</i>). The extension must be .pdf.</p> <p>A data file that matches pattern 2 is 2011-12-02.pdf.</p>
Batching	Overrides patterns	\1\ .oth	The overrides file must be named abc.oth to match data pattern 1.
Batching	File pattern	2011-(\1)\.jdf	The file 2011-12.jdf matches data pattern 2.
	File usage	overrides	
	File type	jdf	
	Required	Yes	
	Sequence	1	
Batching	File pattern	\2\1\ .jdf	The file defabc.jdf matches data pattern 1.
	File usage	overrides	
	File type	jdf	
	Required	Yes	
	Sequence	2	
Batching	File pattern	\1\ .txt	The file 12.txt matches data pattern 2.
	File usage	file	
	File type	txt	
	Required	No	
	Sequence	3	

To edit the Batching tab:

- To add a file pattern, type values in the fields on the **Batching** tab, and click **Add**. Type values in the fields and click **Save**.
- To remove a file pattern, select the checkbox for that file pattern and click **Remove**.
- To edit a file pattern, select the checkbox for that file pattern and click **Edit**. The values display in the file pattern entry fields; change them as needed and click **Save**.
- If you create two file patterns that are the same, or if a pattern's file type and usage match another pattern's file type and usage, a warning message displays, but you can still add the pattern. The first file pattern encountered is processed, based on the value of the **Sequence** property.
- To cancel a change, click **Cancel**. The system cancels the last unsaved change.

Keep these tips in mind as you set up patterns on the **Batching** tab:

- We recommend that you define your patterns carefully, especially for required files, so that only one file matches the specified pattern. It is not possible to specify that more than one file must match a pattern; as soon as one required file matches the pattern, the system considers the requirement met, and will start processing the job when at least one matching required file for each defined pattern is present in the hot folder or SFTP input device.
- Remember to click **OK** to save your changes before leaving the page. If you leave the page without saving your changes, the changes are discarded.
- The rules that you specify in the table are processed in sequence order (from top to bottom); if any conflicting rows exist in your table, the first pattern in the sequence will be used.
- When you use the Pattern batching method, use these guidelines for the fields in the Advanced tab:
 - Use the **Overrides pattern** field in the Advanced tab, not in the Batching tab.
 - Do not use the **JDF patterns** field in the Advanced tab; define those patterns in the Batching tab.

Sets by time

When the batching method is **Sets by time**, the hot folder or SFTP input device submits one or more jobs containing complete sets of files that arrive within a specified time period. The time period is determined by the values set for the **Batching start date**, **Batching start time**, **Batching interval**, and **Batching date** or **Frequency** properties.

For a set to be complete, all of these conditions must be true:

- A file matching the value in the **Data patterns** property is present.
- If a value is entered for the **Overrides patterns** or **JDF patterns** properties, a file matching one or both of those values is present.
- If there are any entries specified for **File patterns**, a file matching each value is present.

The hot folder or SFTP input device waits until the date and time specified in the **Batching start date** and **Batching start time** properties, then submits one or more complete sets of input files at a specific time or time interval determined by the values set for the **Batching interval** and **Batching date** or **Frequency** properties. The hot folder or SFTP input device submits the sets as a group with one set in each child job. If the **Create .zip file** property is set to **Yes**, a single job containing all of the files in all of the sets is submitted in ZIP file format. The files print in the order specified by their **Last modified** timestamp.

Note

- If complete sets of files exist in the hot folder or SFTP input device prior to the initial date and time set in the **Batching start date** and **Batching start time** properties, those sets will be included in the first batch submitted by the input device.
- For the **Batching start time** property, use the time zone of your browser. The value is displayed based on the time zone of the computer that you use to open the user interface, but it is stored in a generic format. The hot folder or SFTP input device interprets the generic time format based on the time zone of its parent server and creates batches at the specified time.

When you use the **Sets by time** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all of the files are combined as a ZIP file and submitted as a single job. If this value is set to **No**, each set is submitted as a child job.
- **Batching start date:** The date when the input device begins to use the **Sets by time** batching method.
- **Batching start time:** The time when the input device begins to use the **Sets by time** batching method.
- **Batching interval:** The time interval used to determine when or how often to submit a group of files.
- **Batching date** or **Frequency:** Used with the **Batching interval** property, this property specifies exact values for when or how often to submit a group of files.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.
- **File Patterns:** A collection of patterns that can define files that are required to be included in a set.
- **JDF patterns:** One or more pattern-matching strings used to identify Job Definition Format (JDF) job tickets. If this property has a value specified, a JDF job ticket is required for the set to be complete.
- **Overrides patterns:** One or more pattern-matching strings used to identify files that contain values for job properties.

Time

When the batching method is **Time**, the hot folder or SFTP input device submits one or more files as a group. Only files that match the **Data patterns** property and that arrive during the time period determined by the values set for the **Batching start date**, **Batching start time**, **Batching interval**, and **Batching date** or **Frequency** properties are submitted.

The hot folder or SFTP input device waits until the date and time specified in the **Batching start date** and **Batching start time** properties, then submits one or more groups of input files at a specific time or time interval determined by the values set for the **Batching interval** and **Batching date** or **Frequency** properties. The files print in the order specified by their **Last modified** timestamp.

Note

- RICOH ProcessDirector for Windows processes child jobs in the order that the files were last modified. The Windows **Date Modified** value is shown as the value of the **Time submitted** job property. Because the **Date Modified** value does not change when the file is placed in the input device, a file created in 2010 and submitted in 2011 shows a **Time submitted** from 2010.

Note

- If files exist in the hot folder or SFTP input device prior to the initial date and time set in the **Batching start date** and **Batching start time** properties, those files will be included in the first batch submitted by the input device.
- For the **Batching start time** property, use the time zone of your browser. The value is displayed based on the time zone of the computer that you use to open the user interface, but it is stored in a generic format. The hot folder or SFTP input device interprets the generic time format based on the time zone of its parent server and creates batches at the specified time.

When you use the **Time** batching method, set these input device properties as indicated:

- **Workflow:** ParentNoPrint
- **Child workflow:** The workflow that is appropriate for the print files in the job.
- **Submit step:** SubmitInputFiles
- **Create .zip file:** If this property is set to **Yes**, all input files that match the value set in the **Data patterns** property are combined as a ZIP file and submitted as a single job. If this value is set to **No**, all input files are submitted as child jobs.
- **Batching start date:** The date when the input device begins to use the **Time** batching method.
- **Batching start time:** The time when the input device begins to use the **Time** batching method.
- **Batching interval:** The time interval used to determine when or how often to submit a group of files.
- **Batching date** or **Frequency:** Used with the **Batching interval** property, this property specifies exact values for when or how often to submit a group of files.
- **Data patterns:** A pattern that matches some part of the file name of your print files. For hot folder or SFTP input devices, the default value is the regular expression: `.*$`. That pattern matches all file names.

Files for batch jobs

Hot folder input devices can submit every file they receive as a separate print job, or they can group print files together to create larger jobs, called **batch jobs**. The simplest batch jobs contain only print files. More complex batch jobs include files that locate the print files and files that set job properties.

Overrides files

When you create a workflow, you can set default values for many job properties. However, those default values might not be appropriate for every job. If you want to send different values with a print job, you can use an overrides file.

An **overrides file** is a text file that contains property and value pairs for job properties; it can be submitted with a print file. When the job is sent to a workflow, the **SetJobPropsFromTextFile** step can use the information in the overrides file to replace the default values.

The overrides file must meet these criteria:

- The overrides file must arrive in the hot folder or SFTP input device **after** the print file.
- You must create a separate overrides file for each print file in the job.
- The file name must match the pattern specified in the **Overrides patterns** property. The default pattern matches files that have the extension **.oth**.
- Each property and value pair must be on a separate line.
- For hot folder or SFTP input devices, each pair must be expressed as: *database property name=value*, with no spaces before or after the equal sign (=).

Note

Overrides files that are used with Download input devices follow a different format and requires the AFP Support feature to be installed.

You can use either of these formats for an overrides file:

- Use the RICOH ProcessDirector database names for the properties, and list each property and value on a separate line. For example:

```
Job.Duplex=Yes
Job.CustomerName=BankName
Job.Location=Boulder
```

- If the AFP Support feature is installed and the overrides arrive in JCL format from a Download input device, the `receive_jcl_jobtype.cfg` file is configured to interpret JCL format. The mainframe program creates the overrides file in JCL format.

To submit one or more print files and their accompanying overrides files in the same job, you must use the **List** batching method and submit a list file with the job. The list file contains the name of each print and overrides file; a print file must be listed before the associated overrides file. The list file might look like this:

```
input1.pdf
values.oth
input2.pdf
morevalues.oth
input3.pdf
values3.oth
inputfile.pdf
inputfile.oth
```

Use the RICOH ProcessDirector database names for the properties, and list each property and value on a separate line. For example:

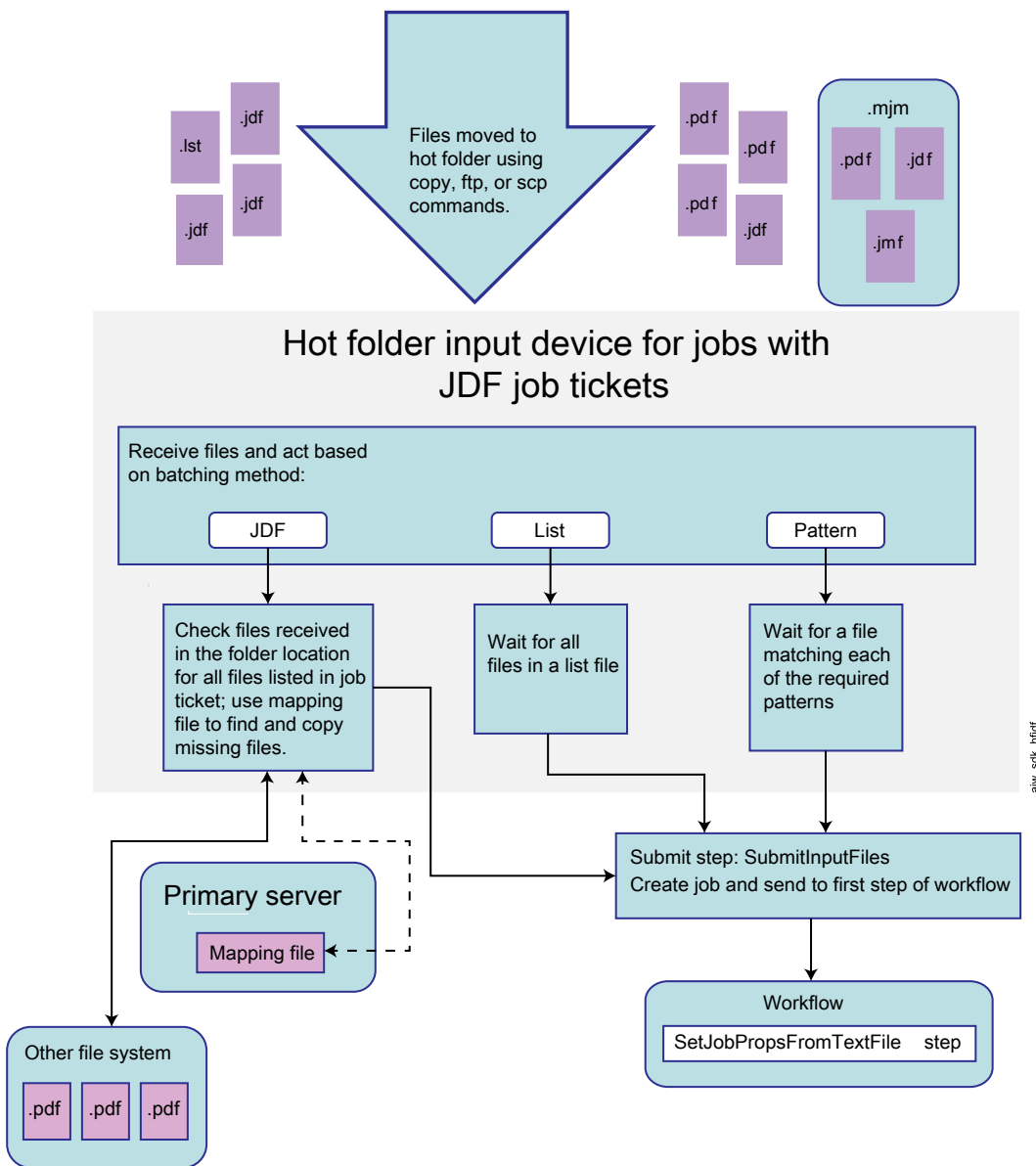
```
Job.Duplex=Yes
Job.CustomerName=BankName
```

JDF job tickets

When you submit a job using a JDF job ticket, the hot folder input device reads the job ticket and uses the contents to find the print files and to set properties for the job. The job is created after all the print files have been found.

The figure [Hot folder input device processing with JDF job tickets, p. 34](#) shows how hot folder input devices process jobs that are submitted with JDF job tickets. The input device accepts job files, processes them based on their batching methods, and sends them to the SubmitInputFiles step.

Hot folder input device processing with JDF job tickets



Setting job properties from the JDF job ticket

RICOH ProcessDirector can use the values that it supports in the JDF job ticket to set job properties.

RICOH ProcessDirector supports functions defined in the JDF Specification that the system requires, including a subset of the JDF Integrated Digital Printing Interoperability Conformance Specification (IDP ICS) and the associated Application Note. The IDP ICS is based on the JDF combined digital printing process, which is intended for integrated digital printers.

RICOH ProcessDirector does not support all possible values in the JDF job ticket. If RICOH ProcessDirector cannot use a value to set a job property, it will remove unsupported values from the job ticket.

↓ Note

- The **IdentifyPDFDocuments** step can receive multiple sets of PDF files and job tickets. The step combines them into a single PDF file and a single job ticket. When the step creates the combined job ticket, it includes only values that RICOH ProcessDirector supports. It does not include unsupported values in the combined job ticket. The **IdentifyPDFDocuments** step is provided by the PDF Document Support feature.

For a list of job properties that can be updated from values in a JDF job ticket, see [Job properties that can be set from the job ticket, p. 189](#).

After the property values are set, RICOH ProcessDirector stores the job ticket with the job files. If the job ticket is requested by a step, RICOH ProcessDirector updates the job ticket with the most current job property values in its database and sends the job ticket to the application called by the step. When the application returns the JDF job ticket, RICOH ProcessDirector updates any supported job property values that have changed, and these can be seen in the job property notebook in RICOH ProcessDirector.

The value of the **Media** property is a special case, because it depends on the setting for **Media Matching**:

- If **Media Matching** is set to **Use media product ID or media name**, RICOH ProcessDirector uses one of these media names as the value of the **Media** property for a job:
 - The name of the media object with the matching product ID specified in the job ticket.
 - The name of the media specified in the job ticket.

RICOH ProcessDirector first checks whether the job ticket specifies a media product ID. If it does, RICOH ProcessDirector looks for a system media object with the same product ID. If RICOH ProcessDirector finds a match, it puts the name of the matching media object in the **Media** property for the job. If RICOH ProcessDirector does not find a match, it looks for a media object with the media name specified in the JDF job ticket. If RICOH ProcessDirector finds a match, it puts the name of the matching media object in the **Media** property for the job.

- If **Media Matching** is set to **Use the properties selected below**, RICOH ProcessDirector uses the media properties (such as size) listed in the job ticket to search the existing system media objects and find one that matches. When it finds an appropriate media object, the name of that object is set as the value of the **Media** property for the job.

You can choose the properties that are used for **Media Matching** based on the needs of your installation.

If more than one media object matches, RICOH ProcessDirector tries to determine which one is the best match based on the rest of the media properties in the job ticket, including the name of the media. If the system cannot determine the best match or if no media objects match, the job goes to Error state. You can use the **Correct Media** action on the job to select the media and move the job out of Error state.

If a job ticket specifies values for media and stapling, you can view and change them in the job properties notebook. On the Scheduling tab, the **Media required** property lists the media values for both the job and any page exceptions. The **Stapling required** property shows whether stapling is required. You can set the job values in the **Media** and **Staple** properties on the **Scheduling** tab. You can change the page values on the **Page Exception** tab using the Page exceptions action.

If a job ticket specifies values for sides exceptions, you can view them using the **Page Exceptions** action on the job. You cannot change the **Sides page exceptions**.

Available punching and stapling options vary from printer to printer. You can configure some RICOH ProcessDirector printers to be punch-capable or staple-capable, but you cannot configure their finishing patterns for punching or stapling. Therefore, RICOH ProcessDirector might schedule a job to a printer that does not support the punch or staple patterns that you requested. When this happens, the printer applies its best equivalent punch or staple pattern.

Viewing the JDF job ticket

Some values in the JDF job ticket are not shown in the RICOH ProcessDirector user interface, but the workflow might call an application that uses these values. Therefore, you might need to view the JDF job ticket.

To view the JDF job ticket:

1. Find the JDF job ticket in the spool file directory for the job.

The spool file directory name is `/aiw/aiw1/spool/default/JobNumber` (AIX and Linux) or `C:\aiw\aiw1\spool\default\JobNumber` (Windows). The name of the JDF job ticket is `JobNumber.filename.jdf`.

If the JDF job ticket has been modified by an overrides file or an application, you might find more than one JDF job ticket with different file names. The date of the last change tells you which JDF job ticket is most recent.

2. Use a text editor or a Web browser to view the JDF job ticket.

The JDF job ticket is in XML format.

Note

RICOH ProcessDirector does not keep the user interface and the JDF job ticket synchronized at all times; it only updates values in the job ticket when it receives a request for the job ticket. When you look at the job ticket, some values listed might not match the values of RICOH ProcessDirector job properties.

Locating files

Applications that submit jobs with JDF job tickets can either send the files along with the job ticket, or leave the files in another location on the network. The job ticket is not required to list the files that the job contains. However, if the job ticket does list the files, it can use either relative path names or absolute path names.

Regardless of how RICOH ProcessDirector finds the files, it copies them into the location specified in the **Folder location** property of the hot folder input device (if they are not there already) and starts to process the job. RICOH ProcessDirector does not automatically delete all the files that are used in jobs that are submitted with JDF job tickets. It only deletes the files in the Folder location. It does not delete files that are in subfolders of the Folder location or that are on other file systems. You must plan to delete the files that are left in the those locations as part of your system maintenance.

Not listing files in JDF

If the job ticket does not list the files, RICOH ProcessDirector expects to receive the job ticket and all the files in the location specified in the **Folder location** property of the hot folder input device. If any of the files are missing, RICOH ProcessDirector continues to wait for them.

In this case, you must use the **List** batching method and submit a list file with the job.

Listing files using relative path names

If the job ticket lists the files using relative path names, RICOH ProcessDirector expects to find all the files in subfolders of the Folder location; they cannot be in parallel folders or on any other file systems. For example, if the job ticket includes the file path `printfiles/test.pdf`, the hot folder expects to find the file in: `[folder location]/printfiles/test.pdf`.

- You can use these formats for a relative path name:

```
dir/filename
filename
```

- These formats are supported but not recommended:

```
file://dir/filename
file://dir\filename
dir\filename
file://./dir/filename
file://.\dir\filename
./dir/filename
.\dir\filename
```

Listing files using absolute path names

If the job ticket lists the files using absolute path names, the files can be anywhere on your network that is accessible from the RICOH ProcessDirector primary server. RICOH ProcessDirector uses the absolute path statements in conjunction with a mapping file to find the files. A **mapping file** is a file that matches the file paths in the job ticket to file paths on mounted file systems. The mapping file is stored on the primary server. Only one mapping file is needed for a primary server; it can list as many file paths as needed.

When RICOH ProcessDirector looks for the files listed in the job ticket with absolute paths, it looks in each location listed in the mapping file until it finds them. If it reaches the end of the mapping file without finding the files, it looks for files received in the Folder location. If it still does not find the files, it waits and checks again at the next poll interval.

- You can use these formats for an absolute path name:

On AIX and Linux:

```
file://hostname/drive:/dir/filename
```

On Windows:

```
file:///drive:/dir/filename
```

Omit `drive:` for files on systems that do not use drive letters.

- These formats are supported but not recommended:

```
file:///drive:\dir\filename
file://drive:/dir/filename
```

```
file://drive:\dir\filename
drive:\dir\filename
drive:/dir/filename
```

- These formats are not supported:

```
file://IP_address/drive:/dir/filename
file://localhost/drive:/dir/filename
```

Note

- You need a file system mapping file if the file paths include identical directory names but have different formats. For example, if the job ticket specifies `file:\\C:\myfiles\testfiles\test1.pdf` and the actual file path is `/myfiles/testfiles/test1.pdf`, you must create a file system mapping file to convert the file path from Windows to Linux or AIX format.
- You need a file system mapping file if the file paths include identical directory names but have different drives. For example, if the job ticket specifies `file:\\D:\myfiles\testfiles\test1.pdf` and the actual file path is `file:\\C:\myfiles\testfiles\test1.pdf`, you must create a file system mapping file to convert the file path from the D drive to the C drive. The mapping looks like this:

```
D:\;C:\
```

Using a mapping file to find input files

If you use a Job Definition Format (JDF) job ticket to submit jobs to a Hot folder device, you do not have to copy all the input files to the Hot folder. When the job ticket is placed in the Hot folder, RICOH ProcessDirector reads it and searches for the input files that it lists.

If the file paths in the job ticket do not match the directory names on a mounted file system, RICOH ProcessDirector uses a file system mapping file to search for the input files.

RICOH ProcessDirector provides a sample file system mapping file, `system_map.cfg`, in `/aiw/aiw1/samples/config/` (AIX and Linux) or `C:\aiw\aiw1\samples\config\` (Windows). You can copy and edit this file as necessary. Copy the file to the `/aiw/aiw1/control_files/config/` (AIX and Linux) or `C:\aiw\aiw1\control_files\config\` (Windows) directory before customizing it. Comments in the sample file explain the file format.

To specify the file system mapping file:

1. Click the **Administration** tab.
2. In the left pane, click **Settings** → **System**.
3. In the **File system mapping file** field, type the file path of the mapping file.
4. Click **SAVE**.

You do not have to restart the system.

List files

You can use list files to process multiple files with or without a JDF job ticket or to process overrides files with files.

The limitations of using a list file with a job ticket include:

- The files must be submitted with the job ticket; they cannot be located on another system and accessed using a mapping file.
- All the files must be submitted directly to the Folder location, not to a subfolder.

If you use a list file, you must use the **List** batching method.

You can use list files in these ways:

- If the job ticket does not include the names of all the input files, the list file can include them along with the name of the job ticket file. When all the files are present, the job is submitted.
- You can create a generic job ticket, which uses placeholder file names instead of real file names, so you can reuse the ticket with different jobs. The list file contains the names of the specific files that are used for the job. The order of the file names in the list file must correspond to the placeholder file names in the job ticket; the first file listed in the list file must correspond to the first placeholder file listed in the job ticket.

For example, if your application prints various booklets that all contain four files (a front cover, a fly leaf, the body of the booklet, and a back cover), you can create a job ticket that includes four files in the order that they should print:

```
frontcover.pdf
flyleaf.pdf
bookletbody.pdf
backcover.pdf
```

Note

In this case, the job ticket must include only file names without any path information.

For each booklet, you can then create a list file that lists the specific files in the same order:

- Booklet 1 list file contains:

```
booklet1fc.pdf
booklet1fl.pdf
booklet1body.pdf
booklet1bc.pdf
```

- Booklet 2 list file contains:

```
booklet2fc.pdf
booklet2fl.pdf
booklet2body.pdf
booklet2bc.pdf
```

- You can create a list file that contains the name of the input file or files and the overrides files that are associated with them. The name of the overrides file must appear immediately after the input file that it is associated with in the list. The names of the input files and overrides files are case-sensitive and each file name must be on a separate line. Do not include directory information with the file name. For example, the list file might contain this information:

```
input1.pdf
prop1.oth
input2.pdf
prop.oth
input3.pdf
duplex.oth
input4.pdf
inputfiles.oth
```

Assigning workflows

When an input file enters the RICOH ProcessDirector system, one of the first actions that the input device does is to assign a workflow to it. After the input device assigns the workflow, the job can begin to move through the processing steps. There are several methods through which the input device can assign the workflow. You configure the method to meet the requirements of the installation.

Use one of these methods to assign the workflow:

- The easiest way to assign a workflow to a job is to use the **Child workflow** property on the input device. When you set the **Child workflow** on an input device, the device assigns that workflow to every job that it processes. All input devices that RICOH ProcessDirector supplies use this method to assign the workflow for the data sets or input files that they receive.


Note

- Jobs that consist of multiple input files are processed as child jobs of the parent job that groups them. The parent job takes the workflow that is set by the **Workflow** property on the input device, while the child jobs take the workflow that is set by the **Child workflow** property on the input device.
- You can set the workflow for a job using a **Child workflow initialization step** on the input device. If you select this method, all jobs that arrive on the input device pass through the initialization step you specify; the initialization step assigns the workflow. If your jobs have multiple input files, also set the **Workflow initialization step** property. Parent jobs pass through that initialization step to have their workflow assigned. RICOH ProcessDirector provides these step templates that you can use for a workflow initialization step:
 - The **SetJobTypeFromRules** step template points to a control file that the input device can use to determine the workflow. The control file lists job parameters that accompany jobs when they are submitted, such as LPR control file parameters, and their corresponding RICOH ProcessDirector properties. For example, you could edit the control file to map the value of the Copies LPR parameter to the value of the **Child workflow** property. You specify the control file that the input device uses with the **Child workflow parsing rules** property on the input device.
 - The **SetJobTypeFromFileName** step template parses the file name of the input file to determine what workflow to assign. To use this step, you specify text that always appears in the name of a certain type of input file and map it to a workflow. Use the **Child workflow pattern** property to define the string you want to use for the workflow.
Make sure that you have created and enabled workflows with names that correspond to the strings you specify. These strings are case-sensitive.
- You can set both the **Child workflow** and the **Child workflow initialization step** properties for an input device. If the **Child workflow initialization step** property is set, the step tries to set the workflow using that method. If the step cannot set the workflow, it assigns the workflow that the **Child workflow** property specifies. If that property has no value, RICOH ProcessDirector issues an error message. You can also set both the **Workflow** and the **Workflow initialization step** properties for an input device to assign the workflow for parent jobs. RICOH ProcessDirector uses the same assignment hierarchy.

Defining a workflow to copy a file to a printer hot folder

You can define a workflow that accepts a print job in a format such as PDF or PostScript or a JDF job ticket and copies it to a hot folder that is associated with a printer.

To define a workflow to copy a file to a printer hot folder:

1. Click the **Workflow** tab.
2. Right-click the **PDF** workflow and select **Copy**.
3. Name the copy of the workflow, fill in or edit other values that you need, and click **Continue**.
4. Right-click the **PrintJobs** step and select **Delete**.
5. If you plan to send PostScript jobs to this workflow, delete the **CountPages** and **CreatePageRanges** steps.
6. Add a **CopyToFolder** step to the **Print** phase
7. Connect the **CopyToFolder** step to the **RetainCompletedJobs** step.
 1. Hover over the edge of the **CopyToFolder** step. Click and hold a highlighted section () to make the connector appear.
 2. Drag the connector onto the **RetainCompletedJobs** step.
8. Connect the **CopyToFolder** step to the step on its left (**CreatePageRanges** for PDF workflows or **RunExternalProgram** for PostScript).
9. Right-click the **CopyToFolder** step and select **Properties**.
10. Click **External**.
11. Delete the contents of the **External Command** property and replace it with one of these commands.

To copy a print file:

- AIX or Linux: `cp ${getCurrentFile(${Job.InputDatastream})} destinationHotFolder/${Job.ID}`
- Windows: `copy ${getCurrentFile(${Job.InputDatastream})} destinationHotFolder\${Job.ID}`

To copy a JDF job ticket:

- AIX or Linux: `cp ${getFileName(overrides,jdf,read)} destinationHotFolder/${Job.ID}`
- Windows: `copy ${getFileName(overrides,jdf,read)} destinationHotFolder\${Job.ID}`

In this text, replace *destinationHotFolder* with the name of the directory that the printer uses as a hot folder.

12. Click **OK**.
13. Save the workflow.

If you also define an AFP printer device to represent the same physical printer, set these properties for the AFP printer device:

- Set the **Share printer connection** property to **Yes**.
- Set the **IPDS printer connection timer** property to a lesser value than the **Inactivity timer** property. If the **IPDS printer connection timer** property is greater than the **Inactivity timer** property, RICOH ProcessDirector drops the connection to the printer before it can share the printer with the hot folder.

Defining a workflow to print to a Passthrough Printer

2

You can define a workflow that accepts a print job in PDF or PostScript format and sends it to a Passthrough printer.

To define a workflow to print to a Passthrough printer:

1. Click the **Workflow** tab.
2. Right-click the **PDF** workflow and select **Copy**.
3. Name the copy of the workflow, fill in or edit other values that you need, and click **Continue**.
4. If the new workflow does not process PDF jobs, right-click the **CreatePageRanges** step, and select **Delete**.

Do not delete this step if the new workflow processes PDF jobs. It is needed to reprint a range of pages in PDF jobs.

5. Connect the **RunExternalProgram** step to the **PrintJobs** step.
6. Right-click the **PrintJobs** step, and select **Properties**.
7. In the **Scheduling** tab, change the values of the **Binding**, **Fold options**, **Requested location**, **Media**, **Output bin**, **Punch**, and **Staple** properties to match the scheduling properties of the Passthrough printer that you want to print on. If you have the AFP Support feature installed, change the values of the **Job class**, **Job form**, and **Job destination** properties as well. Other scheduling properties may be defined in a configuration file.
8. If the workflow processes PDF files and the value of the **Create .zip file** property for the hot folder that assigns jobs to this workflow is set to **Yes**, add a step based on the **BuildPDFFromZIP** step template to the workflow before the first step that expects PDF input (for example, **CreatePageRanges** or **PrintJobs**).
9. Click **OK**.
10. Save the workflow.

Setting up hot folder input devices to process batch jobs

To set up a hot folder input device to process batch jobs, you must configure it to use the appropriate batching method and to recognize the input files.

↓ Note

- All batching methods submit input files when some criteria is met, such as several files are received, a time of day is reached, or a list of expected files arrives. Sometimes, you want to submit the input files before that criteria is met and before the current polling interval ends.

For example, you have one hour until the end of your shift and there are 3857 input files waiting. The input device uses the **Number** batching method and submits jobs when there are 5000 input files waiting. You can use the **Batch all** action for the input device to submit those 3857 files immediately, instead of waiting for more files.

You cannot use the **Batch all** action with the **JDF** or **Pattern** batching method.

To set up a hot folder input device to process batch jobs:

1. Create and save any files that are needed for processing, such as JDF job tickets, list files, or overrides files, or make sure that your process generates these files as needed.
2. Log in to RICOH ProcessDirector.
3. In the **Input Devices** portlet, find the input device that you want to use to process batch jobs.
4. Right-click the input device and select **Properties**.
5. In the left pane, click **Show all tabs** to fully expand the notebook.
6. For any batching method, set these input device properties as follows:

Workflow

ParentNoPrint.

Child workflow

The workflow that is appropriate for the print files in the job.

↓ Note

- If you use the **List** batching method, to be sure that the jobs print in order, select a workflow that includes a step based on the **WaitForGroup** step template before the **PrintJobs** step.
- If you use overrides files, select a workflow that includes a step based on the **SetJobPropsFromTextFile** step template.

Data patterns

One or more pattern-matching strings that define which characters in the data file should be matched with the other file names that might arrive in the input device. If you specify more than one data pattern, the file is considered a match if it matches any pattern. The data pattern is a regular expression.

Submit step

SubmitInputFiles

7. Set these input device properties depending on your batching method:

JDF

JDF patterns

A pattern that matches some part of the name of your job tickets. By default for all input devices except HotFolderJDF this value is null. For HotFolderJDF this value is the regular expression `.*\jdf$`.

List

List patterns

A pattern that matches some part of the file name of your list files. By default, this value is the regular expression `.*lst$`.

Overrides patterns (optional)

A pattern that matches some part of the file name of your overrides files.

Create .zip file

If this property is set to **Yes**, all input files that match the value set in the **Data Patterns** property and that are included in a list file are combined as a .zip file and submitted as a single job. If this property is set to **No**, all input files are submitted as child jobs.

Number and Number of sets

Number of files to batch

The number of files or complete sets that the input device should wait to receive before submitting them as a single group.

Create .zip file

If this property is set to **Yes**, all input files that match the value set in the **Data Patterns** property are combined as a .zip file and submitted as a single job. If this property is set to **No**, all input files are submitted as child jobs. This property is not available for the **Number of sets** batching method.

Pages and Pages in sets

Exceed pages to batch

- **Yes:** A PDF file or complete set of PDF files that contains enough pages to make the total number of pages in the input device exceed the value set for **Number of pages to batch** is submitted with the rest of the files.
- **No:** This file or set remains in the input device as the first set of pages for the next batch.

Number of pages to batch

The number of pages that the input device should wait to receive before submitting the files or complete sets as a single group.

Create .zip file

If this property is set to **Yes**, all input files that match the value set in the **Data Patterns** property are combined as a .zip file and submitted as a single job. If this property is set to **No**, all input files are submitted as child jobs. This property is not available for the **Pages in sets** batching method.

Time and Sets by time

Batching start date

The date when the input device should begin to use this batching method.

Batching start time

The time when the input device should begin to use this batching method.

Batching interval

The time interval that the input device should use to determine when or how often to submit a job.

Batch date or Frequency

Used with the **Batching interval** property, specifies exact values for when or how often to submit a job.

Create .zip file

If this property is set to **Yes**, all input files that match the value set in the **Data Patterns** property are combined as a .zip file and submitted as a single job. If this property is set to **No**, all input files are submitted as child jobs. This property is not available for the **Sets by time** batching method.

Pattern**Data pattern**

One or more pattern-matching strings that define which characters in the data file should be matched with the other file names that might arrive in the input device. If you specify more than one data pattern, the file is considered a match if it matches any pattern. The data pattern is a regular expression.

File pattern (one value for each type of input file)

A pattern-matching string that defines a particular type of input file, for example, `*jdf$` for a job ticket.

Spool file usage (one value for each type of input file)

A value that identifies what the input file is used for, for example, `t i c k e t` for a job ticket or `p r i n t` for a print file.

Spool file type (one value for each type of input file)

The file extension for the input file.

File pattern required (one value for each type of input file)

Whether the job must contain this type of input file.

The job starts to process when all the required files are present. Make sure any optional files are already in the input device before the required files; otherwise they will not be included in the job.

File pattern sequence (one value for each type of input file)

If two or more **File pattern** values are the same, or if two or more **File pattern** properties are associated with the same **Spool file usage** and **Spool file type** values, the order in which the set of values is applied.

8. Click **OK**.

If the input device is enabled and connected, you see a confirmation window asking you if you want to disable and disconnect the input device. To save your changes, the input device must be disabled and disconnected.

9. To use the input device, select it and click **Enable and Connect**.

After you configure an input device to use a batching method, make sure that the input files you submit are appropriate for the batching method selected. Unidentified input files remain in the staging location for the input device with a status of **Waiting**.

Configuring to use JDF job tickets

Before you can submit jobs with JDF tickets, you must configure the input devices that receive the jobs. You also must configure the workflows that the jobs are assigned to. Printers defined as Ricoh PDF printers can manage PDF jobs with JDF tickets better than printers defined as Passthrough or Custom PDF printers can.

If you have Kodak printers that accept PDF, you can send some JDF information to the printer in the KDK data stream recommended by the manufacturer. The Cut Sheet Support for Kodak feature is required.

If you have Xerox printers that accept PDF, you can send some JDF information in the XPIF or XRX data streams recommended by the manufacturer. The Cut Sheet Support for Xerox feature is required.

Before you begin this procedure, review the supplied workflows to see if any of them contain some or all the steps that you want to include. If you find a good workflow, you can copy it and modify it to meet your needs. The workflow must contain a step based on the **SetJobPropsFromTextFile** step template.

Note

- This step template can use an optional `jobID.overrides.jdf` file to set more properties on the job.

In addition, determine whether you can use a hot folder input device that RICOH ProcessDirector provides or whether the installation requires a customized input device. RICOH ProcessDirector provides several hot folder input devices that you can use with only minor modifications or that you can copy to create a customized hot folder input device.

Finally, decide which batching method you want to use. The **JDF**, **List**, and **Pattern** batching methods are all suitable for jobs with JDF job tickets.

To configure to use JDF job tickets:

1. Copy and modify a workflow that contains the processing steps that you want the jobs that are submitted with JDF job tickets to follow:
 1. Click the **Workflow** tab.
 2. Right-click the workflow that you want to copy and select **Copy**.
 3. Name the copy of the workflow, fill in or edit other values that you need, and click **Continue**.
 4. Right-click each step and select **Properties**. Modify the properties as necessary.
 5. Save and enable the workflow by changing , the Save & Enable/Disable switch, to the On position.

6. Repeat these steps if you want to create more workflows.
2. On the RICOH ProcessDirector system, configure an input device so that it assigns the correct workflow for JDF job tickets. We recommend that you copy and rename the supplied HotFolderJDF input device, then verify or update the settings described in the following steps.
 1. Click the **Administration** tab.
 2. In the left pane, click **Devices** → **Input Devices**.
 3. Right-click the HotFolderJDF input device and select **Copy**.
 4. In the left pane, click **Show all** to display all the properties for this input device.
 5. Verify or update the values for these properties:

Workflow

ParentNoPrint

Child workflow

The workflow that is appropriate for the data files in the job.

↓ Note

- If you use the **List** batching method, select a workflow that includes a step based on the **WaitForGroup** step template before the **PrintJobs** step. These steps ensure that the jobs print in order.
- If you use overrides files, select a workflow that includes a step based on the **SetJobPropsFromTextFile** step template.

Data patterns

One or more pattern-matching strings. The strings define the characters in the name of the data file to match with the names of other files that arrive in the hot folder. If you specify more than one data pattern, the file is considered a match if it matches any pattern. The data pattern is a regular expression.

If the value of the **Create .zip file** property is **Yes**, only files that match the data pattern are included in the .zip file.

Submit step

SubmitInputFiles

6. Verify or update the values for these properties, depending on the batching method that you want to use:

Property	Value when the batching method is JDF	Value when the batching method is List	Value when the batching method is Pattern
JDF patterns	A pattern that matches some part of the name of your job tickets. For HotFolderJDF, the default value is the regular expression <code>.*\jdf\$</code> .		
List patterns		A pattern that matches some part of the file name of your list files. By default, this value is the regular expression <code>.*lst\$</code> .	
Overrides patterns	A pattern that matches some part of the file name of your overrides files. For some supplied hot folder input devices, this value is the regular expression <code>.*oth\$</code> .	A pattern that matches some part of the file name of your overrides files. For some supplied hot folder input devices, this value is the regular expression <code>.*oth\$</code> .	
Create .zip file	<ul style="list-style-type: none"> Yes: all input files that match the value set in the Data patterns property are combined as a .zip file and submitted as a single job. If the job ticket specifies different settings for different input files and the Create .zip file property is set to Yes, the settings for the first input file name in the job ticket are 	<ul style="list-style-type: none"> Yes: all input files that match the value set in the Data patterns property are combined as a .zip file and submitted as a single job. If the job ticket specifies different settings for different input files and the Create .zip file property is set to Yes, the settings for the first input file name in the job ticket are 	No

Property	Value when the batching method is JDF	Value when the batching method is List	Value when the batching method is Pattern
	<p>used for all input files.</p> <ul style="list-style-type: none"> • No: all input files are submitted as child jobs. 	<p>used for all input files.</p> <ul style="list-style-type: none"> • No: all input files are submitted as child jobs. 	
File pattern (1 value for each type of input file)			A pattern-matching string that defines a particular type of input file, for example, <code>.*jdf\$</code> for a job ticket or <code>.*pdf\$</code> for a print file.
Spool file usage (1 value for each type of input file)			A value that identifies what the input file is used for, for example, <code>ticket</code> for a job ticket or <code>print</code> for a print file.
Spool file type (1 value for each type of input file)			The file extension for the input file.
File pattern required (1 value for each type of input file)			Whether the job must contain this type of input file.
File pattern sequence (1 value for each type of input file)			If two or more File pattern values are the same, or if two or more File pattern properties are associated with the same Spool file usage and Spool file type values, the order in which the set of values is applied.

3. Make sure that the new input device is connected and enabled.
4. Submit jobs to the new input device. If errors occur, correct the errors that messages from RICOH ProcessDirector or any application that it calls identify.

LPD input devices

If your application submits jobs using the **lpr** client or another command that uses the LPD protocol, the jobs must be sent to an LPD input device. You use the name of the input device as the name of the target printer in the command.

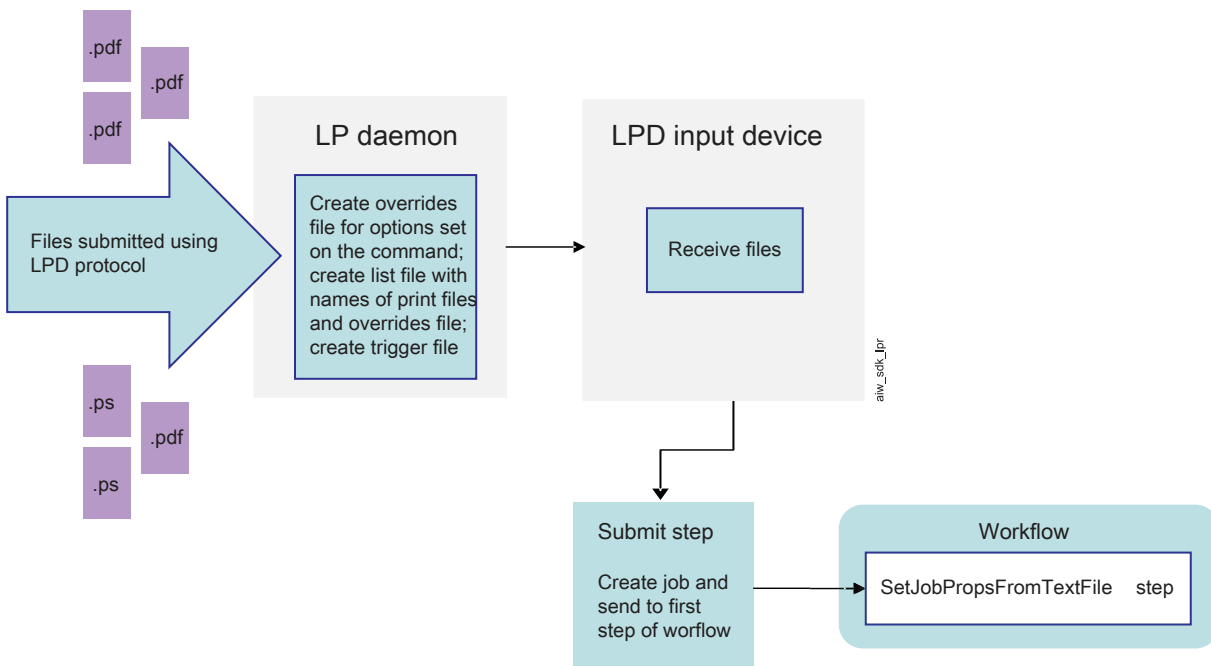
Note

Because some **lpr** commands truncate printer names, we recommend limiting the names of your LPD input devices to 8 characters.

In general, the LPD protocol provides more limited function than other job submission methods. However, some **lpr** clients have more options than others; be sure that you know the capabilities of the **lpr** client that your application uses before you start to configure your input devices.

[LPD input device processing, p. 50](#) shows how the LP daemon and an LPD input device process print jobs. Input files are submitted using the LPD protocol. The LP daemon receives the files and creates an overrides file, a list file, and a trigger file. The daemon sends all the files to the LPD input device, which passes the files to the submit step.

LPD input device processing



Platform differences

The differences between LPD input devices on Linux systems and on AIX systems are:

- On Linux primary and secondary servers, RICOH ProcessDirector installs and uses its own LPD. As a result, you must shut down any other LPDs that are installed; the RICOH ProcessDirector LPD must be the only one running. The RICOH ProcessDirector LPD uses port 515 to receive jobs; no other processes can use that port.
- On AIX primary and secondary servers, RICOH ProcessDirector works with the AIX LPD. When the AIX LPD receives an LPR request, it forwards the request to a component of RICOH ProcessDirector.

Permissions

You can restrict the systems that have permission to submit jobs to LPD input devices by host name or IP address. If your RICOH ProcessDirector system includes Linux primary or secondary servers, you set that list by logging in to RICOH ProcessDirector as an authorized user and updating the **Hosts allowed to submit LPD jobs** system property.

If your RICOH ProcessDirector system includes AIX primary or secondary servers, that property is ignored; the list is maintained in `/etc/hosts.lpd` instead. You can use SMIT to edit `hosts.lpd` on each primary and secondary server. If you update `hosts.lpd`, use these commands to stop and restart the AIX LPD:

```
stopsrc -s lpd
startsrc -s lpd
```

Note

If your RICOH ProcessDirector system includes both AIX and Linux servers, you must update both the system property and `/etc/hosts.lpd`.

LPD options

Generally, `lpr` clients have a limited number of parameters that you can specify. However, if the client you use accepts the `-o` option, you can submit values that can be mapped to RICOH ProcessDirector properties.

LPD input devices on Linux and AIX read the statements that are submitted on the `-o` option and copy them directly into an overrides file. Later in the workflow, a step interprets the overrides file based on the mappings in a control file. RICOH ProcessDirector provides a sample control file, `receive_lpd_overrides.cfg`, in `/aiw/aiw1/samples/rules/` (AIX and Linux) or `C:\aiw\aiw1\samples\rules\` (Windows). You can copy the file and update it to assign values differently if necessary. To modify the file, copy the file to `/aiw/aiw1/control_files/rules/` (AIX and Linux) or `C:\aiw\aiw1\control_files\rules\` (Windows) and make your changes in that directory.

Note

- Updates might overwrite files in the `/aiw/aiw1/samples/` (AIX and Linux) or `C:\aiw\aiw1\samples\` (Windows) directory, but they do not overwrite files in the `/aiw/aiw1/control_files` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory. We recommend copying sample files into the `/aiw/aiw1/control_files` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory and making all your changes in the copied file.

For example, if you need to submit a job to an LPD input device on a Linux server, you can use this command:

```
lpr -S morris -P LPDPDF -O outbin=3 /aiw/aiw1/testfiles/Demo.pdf
```

The LPD input device creates an overrides file that looks like this, with some of the required values filled in from the command and others filled in with default values:

```
orighost=prtroom2.ricoh.com
origuser=root
bannename=/aiw/aiw1/testfiles/Demo.pdf
bannerclass=9.17.160.63
printbanner=Yes
origname=/aiw/aiw1/testfiles/Demo.pdf
outbin=3
```

If the job is submitted to an input device that uses the default control file (`receive_lpd_pdf_jobtype.cfg`), the values are mapped to these properties:

- `Job.Info.NodeID=prtroom2.rioh.com`
- `Job.Host.UserID=root`
- `Job.Name=/aiw/aiw1/testfiles/Demo.pdf`
- `Job.OutputBin=3`

The values for **bannerclass** and **printbanner** are left unmapped because they do not appear in the default control file.

Processing flow

LPD input devices always have these settings; you cannot change them:

- **Batching method:** List
- **Completion method:** Trigger
- **Data patterns:** `.*\prt$`
The lp daemon adds the suffix **.prt** to all the print files it receives, so this pattern always matches. The names of your print files do not have to include those characters.
- **List patterns:** `.*list\lst$`
- **Overrides patterns:** `.*other\oth$`
- **Trigger patterns:** `.*\trg$`

When a job is submitted to an LPD input device, the LPD receives the print file and any options (flags) that were set on the command. The LPD creates an overrides file for the job and writes the options into it, in a format that can be interpreted by a later step. If the `lpr` command supports the **-o** option for including additional job properties, those properties are also written into the overrides file. The file is saved with the extension **.oth**.

The LPD also creates a list file for the job. The list file includes the names of the print file (or print files, if multiple files are submitted) and the overrides file, and has the extension **.lst**. The names of print files are added to the list file in the order they are received. Although some `lpr` clients send files in the order they are listed in the command, not all of them do. Files might arrive in random order.

When the LPD finishes creating the overrides and list files, it creates a trigger file. A trigger file does not contain any data; its presence indicates to the input device that the input file is ready to be processed. When the **Completion method** of an input device is set to **Trigger**, the input device waits until it detects a file that matches one of the **Trigger patterns** set on the device.

After the input device finds the trigger file, it submits the job to the first step of the workflow for processing. That step should be based on the **SetJobPropsFromTextFile** step template. The step uses the control file that the input device specifies specified in the **Child workflow parsing rules** property to interpret the overrides file and assign values to job properties. The default control file is `receive_lpd_jobtype.cfg` or `receive_lpd_pdf_jobtype.cfg`.

Configuring to use the LPD protocol

Before a user can use the line printer daemon (LPD) protocol to submit jobs to the RICOH ProcessDirector system, an administrator must do configuration tasks on the RICOH ProcessDirector system to configure the input devices that receive the jobs, to configure the workflows that the jobs are assigned to, and to define the hosts that can use the LPD protocol to submit jobs to RICOH ProcessDirector. The administrator might also have to do some configuration tasks on these hosts.

Before you begin this procedure, review the supplied workflows. If you find one that contains some or all the steps that you want to include in your workflow, you can copy it and modify it to meet your needs.

In addition, determine whether you can use one of the LPD input devices that RICOH ProcessDirector provides or whether the installation requires a customized input device. RICOH ProcessDirector provides LPD input devices that you can use with only minor modifications or that you can copy to create a customized LPD input device.

To configure to use the LPD protocol:

1. If any LPD daemons or processes that do not belong to RICOH ProcessDirector (for example, the Common UNIX Printing System [CUPS] LPD daemon) are running on the same system as the parent server for the input device, stop them.
 - On SUSE Linux, start YaST, then click **System** → **Network Services** to find the LPD services.
 - On Red Hat 6.7 through latest 6.X, choose **System** → **Administration** → **Services** to find the LPD services.
 - On Red Hat 7.1 through latest 7.X, choose **Applications** → **Utilities** → **Terminal** and use this command: `systemctl list-units -type service -all`.
To stop an LPD service, use this command: `systemctl stop <LPD_service>`.
 - On Windows, to stop the LPD service or the TCP/IP print server, go to the Control Panel and double-click **Services**. Select the service and click **Stop**. To stop other LPD processes, see the documentation for your LPD product.

★ Important

- Do not uninstall CUPS.
2. If the parent server of the input device is defined on a Linux computer or a Windows primary computer, update the system setting to specify the hosts that can submit jobs using the LPD protocol.

If the base product is installed on AIX, you only have to do this step if the parent server of the input device that receives jobs from LPD is a Linux secondary server.

1. Click the **Administration** tab.
2. In the left pane, click **System** → **Settings**.
3. In the **Hosts allowed to submit LPD jobs** field, type the allowed host names or IP addresses.

Separate host names and IP addresses with semicolons.

↓ **Note**

- You can use wild cards in host names and IP addresses (for example, *.acmeproducts.com or 192.*). A value of * means that all hosts are allowed to submit jobs. Values that contain only numbers, decimal points, and wild cards are compared to IP addresses. Values that contain wild cards and at least one alphabetic character are compared to host names. An empty value means that no hosts are allowed to submit jobs.
- The default value is *.
- If the base product is installed on AIX or Linux, you cannot submit jobs to LPD input devices from the primary computer.
- If you experience long wait times or missing jobs, set the LPD host entries to IP addresses or fully qualified host names (such as hostserver.co.acmeproducts.com instead of *.acmeproducts.com).

4. Click **OK**.

3. If the parent server of the input device is defined on a Linux computer or a Windows primary computer, update the system setting to specify the hosts that can submit jobs using the LPD protocol.

If the base product is installed on AIX, you only have to do this step if the parent server of the input device that receives jobs from LPD is a Linux secondary server.

1. Click the **Administration** tab.
2. In the left pane, click **Settings** → **System**.
3. In the **Hosts allowed to submit LPD jobs** field, type the allowed host names or IP addresses.

Separate host names and IP addresses with semicolons.

↓ **Note**

- You can use wild cards in host names and IP addresses (for example, *.acmeproducts.com or 192.*). A value of * means that all hosts are allowed to submit jobs. Values that contain only numbers, decimal points, and wild cards are compared to IP addresses. Values that contain wild cards and at least one alphabetic character are compared to host names. An empty value means that no hosts are allowed to submit jobs.
- The default value is *.
- If the base product is installed on AIX or Linux, you cannot submit jobs to LPD input devices from the primary computer.
- If you experience long wait times or missing jobs, set the LPD host entries to IP addresses or fully qualified host names (such as hostserver.co.acmeproducts.com instead of *.acmeproducts.com).

4. Click **SAVE**.

4. If the parent server of the input device is defined on an AIX computer, make sure the computer is configured to accept jobs submitted using the LPD protocol.

If the base product is installed on Linux, you only have to do this step if the parent server of the input device that receives jobs from LPD is an AIX secondary server.

1. On the AIX computer, log in as the root user.

2. Make sure that the LPD daemon is installed and running.

To start the LPD daemon using the System Management Interface Tool (SMIT), click **Print Spooling** → **AIX Print Spooling** → **Manage Print Server** → **Start the Print Server Subsystem (LPD daemon)**.

3. Add the host name or IP address of the systems that you want to allow to submit jobs to the /etc/hosts.lpd file:

- 1) In SMIT, click **Print Spooling** → **AIX Print Spooling** → **Manage Print Server** → **Add Print Access for a Remote Client**.

- 2) In the **Name of REMOTE CLIENT** field, type the host name or IP address of the system that you want to allow to send jobs to the RICOH ProcessDirector server.

↓ **Note**

- If you are not using Domain Name Services (DNS), you must fully qualify the host name. For example: mathomhouse.micheldelving.org.

- 3) Click **Enter**.

5. On each system that you authorized to submit LPD jobs, determine if the print command lets you specify a server name. If it does not, do one of the following steps to create a print queue on the system to send jobs to the LPD input device.

↓ **Note**

- The LPR client that is supplied with the base operating system in some versions of Windows and that is available as an optional feature of other versions lets you specify a server name. If you use this LPR client or an equivalent, you do not have to create a print queue on Windows.

6. To create a print queue on SLES 12.0:

1. Log in as the root user.

2. Start YaST.

3. Click **Hardware** → **Printer**. With **Printer Configurations** highlighted, click **Add**. Click **Connection Wizard**, and then select **Line Printer Daemon (LPD) Protocol**.

4. In the **IP Address or Host Name** field, type the host name or IP address of the system where the LPD input device is defined.

5. Type the name of the LPD input device in the **Queue Name** field, and click **OK**.

6. In the **Set Arbitrary Name** field, type the name of the LPD input device. This name must be unique on this Linux system. Although LPD input device names are case-sensitive, Linux does not allow you to define multiple LPD input device names that are alike except for case. For example, you cannot define one LPD input device called HotFolderLPD and another called hotfolderlpd.

7. Click **OK**.

7. To create a print queue on Red Hat 6.7 through latest 6.X:

1. Log in to the operating system as the root user.

2. From the Gnome Desktop, click **Applications** → **System Tools** → **Terminal**.

3. In the terminal window, type `system-config-printer`.


4. Select **Printer** on the **New** drop-down menu.
 5. Click **Network Printer** and select **LPD/LPR Host or Printer**.
 6. In the **Host** field, type the host name or IP address of the system where the LPD input device is defined.
 7. Type the name of the LPD input device in the **Queue** field, and then click **Forward**.
 8. In the **New Printer** window, click **Forward**.
 9. Click **Forward** again.
 10. Type a name for the LPD input device in the **Printer Name** field.
 11. Click **Apply**.
8. To create a print queue on Red Hat 7.1 through latest 7.X:
 1. Log in to the operating system as the root user.
 2. From the Gnome Desktop, click **Applications** → **Utilities** → **Terminal**.
 3. In the terminal window, type `system-config-printer`.
 4. Select **Printer** on the **Add** drop-down menu.
 5. Click **Network Printer** and select **LPD/LPR Host or Printer**.
 6. In the **Host** field, type the host name or IP address of the system where the LPD input device is defined.
 7. Type the name of the LPD input device in the **Queue** field, and then click **Forward**.
 8. In the **New Printer** window, click **Forward**.
 9. Click **Forward** again.
 10. Type a name for the LPD input device in the **Printer Name** field.
 11. Click **Apply**.
 9. To create a print queue on AIX:
 1. Log in as the root user.
 2. Start SMIT.
 3. Click **Print Spooling** → **AIX Print Spooling** → **Add a Print Queue** → **Remote** → **Standard processing**.
 4. In the **Name of QUEUE to add** field, type the name of the LPD input device.
 5. In the **HOSTNAME of remote server** field, type the host name or IP address of the system where the LPD input device is defined.
 6. In the **Name of QUEUE on remote server** field, type the name of the LPD input device.
 10. On the RICOH ProcessDirector system, copy and modify a workflow that contains the processing steps that you want the jobs that are submitted using the LPD protocol to follow.

To copy and modify one or more workflows:

1. Click the **Workflow** tab.
2. Right-click the workflow that you want to copy, and click **Copy**.
3. Name the copy of the workflow, fill in or edit other values that you need, and click **Continue**.

4. Right-click each step and select **Properties**. Modify the properties as necessary.
Remove `${Job.InputFile}` from the **Job name** property in the **SetJobPropsFromTextFile** step.
5. If you have the AFP Support feature installed and the AFP resources (such as fonts, overlays, and page segments) required by the jobs that are processed through this workflow are not going to be sent inline with the input file, make sure that those resources are available to the RICOH ProcessDirector system. We recommend that you move these resources to `/aiw/aiw1/resources` (AIX and Linux) or `C:\aiw\aiw1\resources` (Windows) or `/usr/lpp/psf/reslib` (AIX and Linux) or `C:\Program Files (x86)\Ricoh\PSF\reslib` (Windows), so that they are available to all the components of RICOH ProcessDirector. If you cannot use those directories, you can set the **AFP resource path** property on one of the steps in the workflow to refer to the directory or directories that hold the resources.

Note

- The **AFP resource path** can be set as a default job property on various step templates, including **EnableRepositioning**, **PrintJobs**, and **ConvertLineDataJobIntoAFP**. You only need to set the value on one of the steps; the others inherit the value.
6. To use the workflow, save and enable it by changing , the Save & Enable/Disable switch, to the On position.
 7. Repeat these steps if you want to create additional workflows.
11. On the RICOH ProcessDirector system, configure an input device so that it assigns the correct workflow or workflows for the input files that it receives. We recommend that you copy and rename one of the supplied LPD input devices, then verify or update the settings described below.
 1. Click the **Administration** tab.
 2. In the left pane, click **Devices** → **Input Devices**.
 3. Right-click the input device that you want to copy and select **Copy**.

Note

- The new input device is the same type as the copied input device. For example, you cannot create a new LPD input device by copying a hot folder.
4. In the left pane, click **Show all tabs** to display all the properties for this input device.
 5. Verify or update the values for these properties:

Input device name

Make sure that the input device name does not include any spaces. The LPR client cannot process names with spaces.

It is best to limit the input device name to 8 characters. Depending on the print command that you use, you might have to create a print queue on the sending system with the same name as the input device. Some systems truncate print queue names to 8 characters.

Folder location

The directory on the primary computer that receives jobs from authorized hosts. Make sure that the file system is set up so that the directory you list here is large enough to handle the amount of data that the LPR client sends without filling the file system.

Staging location

The directory that RICOH ProcessDirector moves input files to before they are submitted as jobs. Make sure that the file system is set up so that the directory you list here is large enough to handle the amount of data that the LPR client sends without filling the file system. Remember that there might be two copies of an input file in the system at any time, one in the **Folder location** directory and one in the **Staging location** directory.

Parent server

The RICOH ProcessDirector server where the files will be received; for example, a submitter would specify this server name on the **lpr** or **lprafp** command. The server specified here must be configured to accept jobs over the LPD protocol.

- To assign workflows to jobs, with either single or multiple input files:

↓ Note

- Input devices with a Linux parent server cannot create a parent job with multiple children. Instead, one parent job and one child job are created for each input file.
- Set the **Submit step** property to **SubmitInputFiles** and the **Workflow** property to **ParentNoPrint**.
 - Determine how you want the input device to assign the workflow for each single job or child job. You can select one of these:

- Set the **Child workflow initialization step** property to **SetJobTypeFromRules** and use the **Child workflow parsing rules** property to specify the name of the control file that can set the workflow from a value of an option of the print command.
RICOH ProcessDirector provides two sample control files that are used to set the workflow. The sample control files, called `receive_lpd_jobtype.cfg` and `receive_lpd_pdf_jobtype.cfg`, are installed in the `/aiw/aiw1/samples/rules/` (AIX and Linux) or `C:\aiw\aiw1\samples\rules\` (Windows) directory. You can copy one of the files to the `/aiw/aiw1/control_files/rules/` (AIX and Linux) or `C:\aiw\aiw1\control_files\rules\` (Windows) directory and modify it to meet your needs, then update the value of the **Child workflow parsing rules** property to point to your file.
- Set the **Child workflow initialization step** property to **SetJobTypeFromFileName** and use the **Child workflow pattern** property to specify the string that RICOH ProcessDirector should look for in the input file name and use as the workflow name. If you use this method, you must make sure that a workflow with the corresponding name exists.

- If the control file that you created in the previous step sets job scheduling properties such as **Media** or **Job size** based on the values of options of the print command, make sure that the corresponding scheduling properties are set on the target printers in RICOH ProcessDirector. If the scheduling properties do not match, the jobs are not automatically scheduled to those printers.
- Make sure that the LPD input devices are connected and enabled.

↓ Note

- The LPD input device does not return status information in response to the **lpq** command.
- From a host system that you authorized, submit some test jobs. If errors occur, correct the errors that messages from the host system or RICOH ProcessDirector identify.

Print commands used with the LPD protocol



The LPD protocol defines a print command called **lpr**. You can use the **lpr** command, or several other commands that use the LPD protocol, to submit jobs to LPD input devices.

Syntax and available options vary from one print command to another and from one version of **lpr** to another. You might have to adapt examples of print commands for your own system.


You might also have to configure the sending system to compensate for options that your print command does not provide. For example, if the print command does not have an option for specifying a print server, you must create a remote print queue on the sending system with the same name as the LPD input device.

Here are a few common LPD print commands. For more information about these commands, see the operating system or product documentation.



Linux print commands

Command name	Where to get it	Printer option	Server option	Job properties option
lpr	Native	-P	None	Varies  Note Some LPR clients let you send job options with the -o command. See the documentation for your LPR client to determine if you can do this.
lprafp	Download from the Ricoh Production Print Web site (downloads for InfoPrint Manager for Windows)	-P	-s	-o  Note If you submit jobs using lprafp and include options on the -o flag, you must use only single-byte characters in the options. If you use double-byte characters in the options, RICOH ProcessDirector cannot read them and the lprafp request is rejected.

AIX print commands

Command name	Where to get it	Printer option	Server option	Job properties option
lpr	Native	-P	None	None
enq	Native	-P	None	-o
lp	Native	-d	None	-o
lprafp	InfoPrint Manager for AIX; download latest version from the Ricoh Production Print Web site (downloads for InfoPrint Manager for Windows)	-P	-s	<p>-o</p> <p> Note</p> <p>If you submit jobs using lprafp and include options on the -o flag, you must use only single-byte characters in the options. If you use double-byte characters in the options, RICOH ProcessDirector cannot read them and the lprafp request is rejected.</p>
qprt	Native	-P	None	None
qprt	InfoPrint Manager for AIX	-P	None	-o

Windows print commands

Command name	Where to get it	Printer option	Server option	Job properties option
lpr	Native	-P	-S	None  Note The Windows lpr command has an -o option, but it indicates the file type, not job properties.
lprafp	Download from the Ricoh Production Print Web site (downloads for InfoPrint Manager for Windows)	-P	-s	-o  Note If you submit jobs using lprafp and include options on the -o flag, you must use only single-byte characters in the options. If you use double-byte characters in the options, RICOH ProcessDirector cannot read them and the lprafp request is rejected.

Examples

This command submits a file called `report.pdf` from a Windows system to an LPD input device called LPDPDF that is defined on a server called morris and requests two copies:

```
lpr -S morris -P LPDPDF -# 2 report.pdf
```

Either of these commands submits a file called `report.afp` from an AIX system to an LPD input device called LPDAFP that is defined on a server called morris. To use the AIX **lpr** command, you must first define a remote print queue for LPDAFP on the AIX system that you are sending the job from.

```
lpr -P LPDAFP -C A report.afp
lprafp -smorris -pLPDAFP -oclass=A report.afp
```

Restricting use of the LPD protocol to submit jobs

You can limit the hosts that can use the LPD protocol to submit jobs to all the input devices with the same parent server.

To restrict the use of the LPD protocol:

- If the parent server of the input devices is a Linux computer or Windows primary computer, change the **Hosts allowed to submit LPD jobs** system property.

The default is to allow input from all systems.

1. Click the **Administration** tab.
2. In the left pane, click **Settings** → **System**.
3. In the **Hosts allowed to submit LPD jobs** field, type a list of authorized host names or IP addresses separated with semicolons.

You can use an asterisk (*) to represent zero or more characters. Values that contain only digits (0–9), decimal points (.), and asterisks (*) are compared to the IP address of the incoming connection. Values that contain one or more alphabetic characters (A–Z, a-z) are compared to the host name of the incoming connection.

A list value of * means that all hosts are allowed to submit jobs. An empty list value means that no hosts are allowed to submit jobs.

- If the parent server of the input devices is an AIX computer, edit the /etc/hosts.lpd file on the AIX computer.

The default is to deny input from all systems.

1. Log in to the operating system as the root user.
2. Start SMIT.
3. Click **Print Spooling** → **AIX Print Spooling** → **Manage Print Server** → **Add Print Access for a Remote Client**.
4. In the **Name of REMOTE CLIENT** field, type the host name or IP address of the system that you want to allow to send jobs to the RICOH ProcessDirector server.

 **Note**

If you are not using Domain Name Services (DNS), you must fully qualify the host name. For example: mathomhouse.micheldelving.org.

5. Press Enter.

3. Adding functions to workflows

- **Setting up external programs**
- **External steps**
- **Setting up step templates for external steps**
- **Setting up workflows for external steps**

RICOH ProcessDirector workflows can be configured to send print jobs to other applications for additional processing and then receive them back into the system. The other applications can run on the server that holds the primary server, on a secondary server, or on an application server.

If your application provides processing that can be incorporated into print workflows (such as address cleansing or impositioning), you can create steps that send the print job from RICOH ProcessDirector to your application, wait for the application to finish, and then continue processing in RICOH ProcessDirector. The RICOH ProcessDirector documentation calls these steps **external steps**.

In a conditional workflow, external steps can receive input from multiple steps and run different programs based on the input. In the **PDFProduction** sample workflow, the external step receives different properties based on a job's page count. External steps can write results to multiple places and send output to multiple steps. For example, an external step can send jobs to three different steps based on a value computed by an external program. An external step can terminate a branch of a conditional workflow.

These step templates for external steps are included with RICOH ProcessDirector:

- **RunExternalProgram**
Submits the print job to an external program using a command line interface.
- **RunHotFolderApplication**
Puts the print job in a hot folder that another application monitors. That application picks up the print job, processes it according to its configuration settings, and puts it into an output folder. RICOH ProcessDirector monitors that output folder and retrieves the job when it is ready.

Choosing which step template to use depends primarily on how the external application can accept jobs. If the application has a command line interface, the workflow can include **RunExternalProgram**; if the application monitors hot folders for input, the workflow can include **RunHotFolderApplication**. If the application has both interfaces, you can decide which step to use.

RICOH ProcessDirector provides optional features that integrate with external programs. These features include additional step templates that you can use:

- **Quadient Inspire Connect**
 - **ComposePDF**
Uses Quadient Inspire Designer to generate a new PDF file from a WFD file and one or more raw data files. The step waits for the new file to be returned.
 - **ComposeAFP**
Uses Quadient Inspire Designer to generate a new AFP file from a WFD file and one or more raw data files. The step waits for the new file to be returned. **ComposeAFP** is only available when the Quadient Inspire Connect and AFP Support features are installed.
- **MarcomCentral Connect**
 - **ApplyXSLTransform**
Converts information in MarcomCentral format to RICOH ProcessDirector job properties.

- **CallSOAPService**
Calls the MarcomCentral **Job Tickets** web service and retrieves the job tickets for each order.
- **CreateJobsFromXML**
Uses the information returned by the MarcomCentral web service to create a job for each job ticket. The step submits the jobs to the **MarcomProcessJobTicket** workflow.
- Pitstop Connect
 - **RunPitStopOnJob**
Submits a PDF print job to PitStop Server along with an action list or PDF profile. Ricoh ProcessDirector waits for PitStop Server to process and return the job so it can move to the next step in the workflow.
- Postal Enablement
 - **SetPostalJobProps**
Lets you set job properties needed by postal software to determine how to process the mail piece information contained within the external document properties file.
 - **BuildExternalDocPropsFile**
Lets you extract document data from the document properties file and create a file with the document data and headings that you need to send to an external program. The file you create is called an external document properties file.
 - **MapExternalResultsFiletoDocProps**
Maps document properties that you select from an external results file to the document properties included in a modified results file. The external results file is produced by an external program.
 - **UpdateDocPropsFromExternalResultsFile**
Merges the properties in the modified results file into the document properties file for the job.
- Ultimate Impostrip® Connect
 - **RunImpostripOnJob**
Submits a PDF print job to the Ultimate Impostrip® input hot folder that is set up to perform the appropriate imposition functions on the job. RICOH ProcessDirector waits for the Ultimate Impostrip® server to return the updated job, then continues processing with the next step in the workflow.

Consider these points when deciding which step to use:

- **RunExternalProgram**
 - Requires you to write a command and specify all of the necessary options. If you are not familiar with writing commands or do not have the syntax of the command to run the application, this step might be more difficult to configure.

 **Note**

All output that external programs write to stdout and stderr appears in the RICOH ProcessDirector job log. If your command runs a program or system command that writes all its output to stdout, we recommend that you use a script to call the external program so you can redirect the output to a different location.

- Lets you specify the code page and language to be sent to the other application.
- Lets you specify the return codes that the program can provide without sending the print job into an error state.
- Lets you use a control file that contains processing parameters that the external application can use, if the program can accept one.

- **RunHotFolderApplication**

- Lets you send one or more files to the other application.
- Requires you to configure the sending and retrieval folders correctly. The sending folder and retrieval folder can be the same one, if the file name of the file to send does not match the retrieval pattern. The **Retrieval pattern** is a pattern-matching string used to identify a returned job.

For example, if the external application converts files from PostScript to PDF, the retrieval pattern could look for a file with the extension `.PDF`. The step can copy `printfile.ps` into the directory and not mistake it for the converted file, because it is looking for `printfile.PDF`.

However, if you use the hot folder to send a PDF file to a preflight application, that application sends back another PDF file, most likely with the same name. In that case, the sending and retrieving folders must be different locations.

- Does not provide the option to specify valid return codes.

If the external application returns the expected file, the step picks up the file and moves the process to the next step without checking for errors. As a result, the other application should be configured so it does not copy a file with errors to the output directory. If the hot folder is set to time out, the job eventually goes into an error state and the issue can be resolved; if the hot folder is configured to wait indefinitely, the job continues to wait until an operator intervenes.

In some environments, the steps can be configured to work together. For example, if you want to use FTP or SCP to transfer the input file to the other program, you can create a step based on **RunExternalProgram** to transfer the files. Then, you can add a step based on **RunHotFolderApplication**, leaving the **Sending folder** property blank. The step does not send any files to an outgoing hot folder, but it polls the retrieving folder for the resulting file.

Processing flow: RunExternalProgram

Steps based on the **RunExternalProgram** step template can access files in the spool directory for a job. When an input device creates a job from an input file, it also creates a unique spool directory for the job. The spool directory contains a copy of the input file and other files that provide information about the job. External steps can call programs that update existing files in the spool directory and write new files to the directory.

When you create a step based on the **RunExternalProgram** step template, you can include values for these properties:

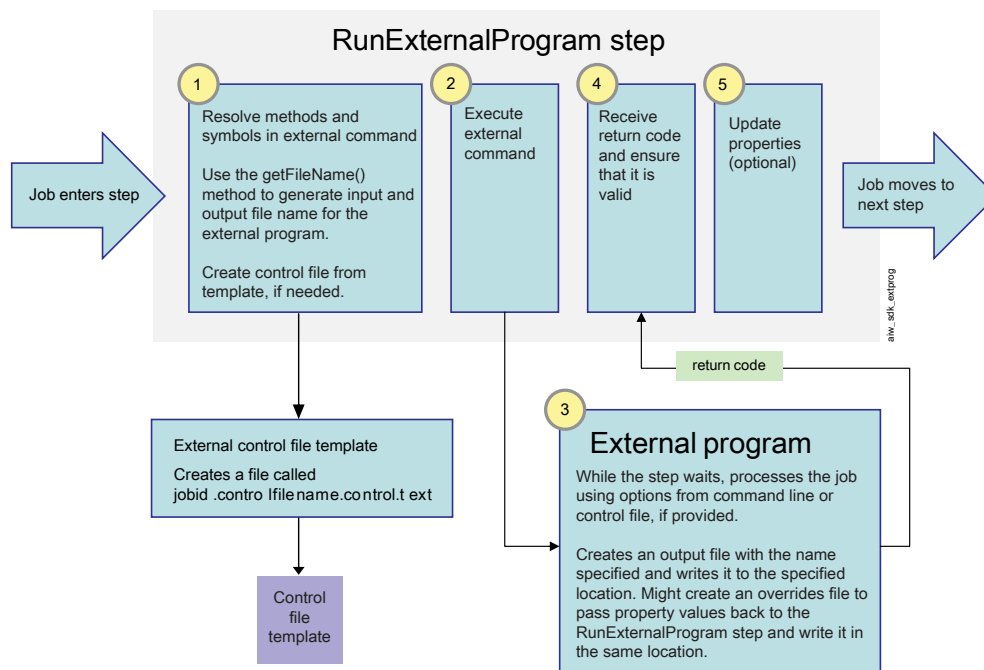
- External command
- External control file template

- External program code page
- External program language
- Valid return codes

The next figure shows how the `RunExternalProgramStep` step processes a job. The step resolves methods and symbols in the command, creates a control file if needed, runs the command, receives a return code when the other program finishes processing, and updates job properties if the program sends an overrides file back. Then the job moves to the next step.

RunExternalStep step template processing

3



When a job enters the **RunExternalProgram** step, the step tries to run the command provided in the **External command** property. If any part of the command is written in RICOH ProcessDirector symbol notation, the step resolves all references before it issues the command:

- The command must contain two file names: the input file name (the file that the other program picks up) and the output file name (the file that the other program returns to RICOH ProcessDirector). The step must generate these names using the RICOH ProcessDirector method `getFileName()`. Instead of using actual file names in the command, the method is used as a symbol for the file names.

For example, the input file name might be represented by the symbol: `#{getFileName(print, pdf, read)}` and the output file name might be represented by the symbol: `#{getFileName`

(print, pdf, write)}. Using this method ensures that the application can find and return the files correctly.

- The command can include symbols for RICOH ProcessDirector properties; if it does, those property values are entered into the command.
For example, if the other application needs to know the job name, you can include the symbol: `${Job.Name}`. The step finds the value of the **Job name** property and includes it in the command.
- The application might accept property values in a parameter or control file.
If it does, the command should include the **getControlFileName()** method to create a name for the control file. The step uses the control file template listed in the **External control file template** property to build the contents of the control file, then inserts the file name in the command.
- The application might accept values in a JDF job ticket.
If it does, the command should include the **getFileName()** method to generate the name of the JDF job ticket. An example of this syntax is `${getFileName(overrides, jdf, read)}`.

After all of the symbols have been resolved, the command runs.

The external application does its processing, copies its output files to the correct locations, and returns a return code. The output files can include:

- The modified print file, named based on the value of the **getFileName()** symbol that was included in the command
- A file used to pass property values back into RICOH ProcessDirector
RICOH ProcessDirector that can accept property values from external programs in these ways:
 - In a file named *jobid.overrides.txt* or *.text*, which contains a list of RICOH ProcessDirector attribute/value pairs
 - In an empty file with a file name that includes the attribute/value pairs, which is created in the spool file location if the file contains property value pairs on the output parameter of the command

The step compares the return code to the values listed in the **Valid return codes** property for the step. If it matches one of the codes, processing continues; if it does not match, the job goes into an error state.

The step looks for the file that carries the property values. If it finds a file, it reads the file (or file name) and updates the property values. Then the step completes its processing. It might pass the job to another step or terminate a branch of the workflow.

Processing flow: RunHotFolderApplication

When you create this step, you can include values for these properties:

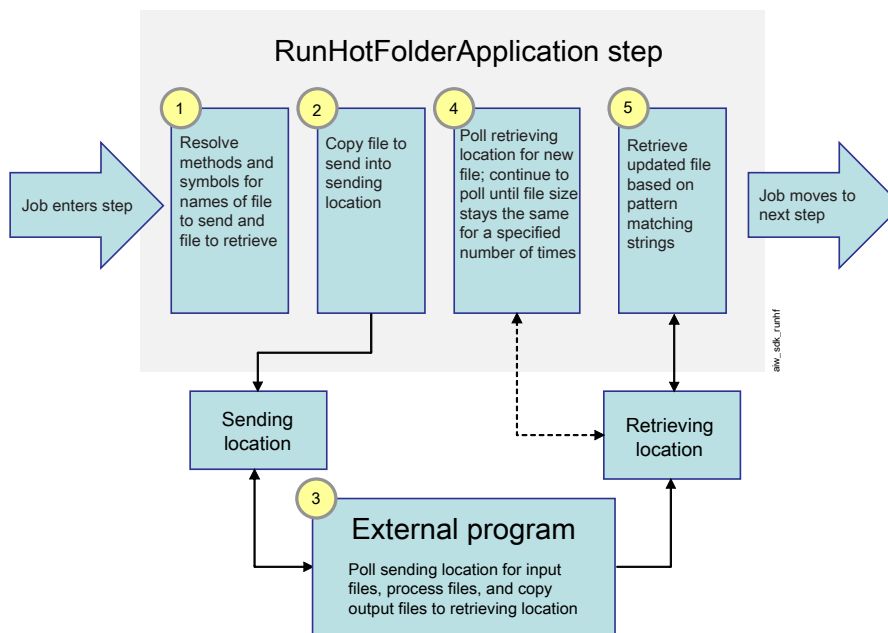
- File size verification count
- File to send
- Poll interval
- Retrieval folder
- Retrieval pattern
- Retrieved file
- Sending folder

- Timeout interval

The next figure shows how the RunHotFolderApplication step processes a job. The step resolves the method and symbol for the name of the file to send, copies the file into the sending location, then waits as the external program processes the file. The step monitors the retrieving location for the file and retrieves the file when it has been fully copied into the location. Then the job moves to the next step.

The step can process jobs that have a single file or multiple files.

RunHotFolderApplication step template processing



When a job enters the step, the step resolves the value of the **File to send** property and locates that file in the pool directory. The step copies the print file into the location specified in the **Sending folder** property. The step expects the other application to pick the file up from that location.

After the file is copied, the step starts to monitor the directory listed in the **Retrieval folder** property. The step looks in the folder for a file whose name matches the pattern specified in the **Retrieval pattern** directory. If no file that meets the criteria is present, the step waits for the amount of time specified in the **Polling interval** property and checks again.

The step continues to check for the file for a given period of time. This length of time is determined by the value of the **Timeout interval** property. The timeout interval is calculated from the time the print file is copied into the **Sending folder**. If the modified file does not appear in the **Retrieval folder** before the timeout interval elapses, the job goes into an error state.

When the step finds the expected file in the directory for the first time, it notes the size of the file, but does nothing else. The step must make sure that the entire file has been transferred from the other application before it continues processing. After the polling interval has elapsed, the step checks again, finds the file, and compares the file size to the previous size.

- If the file is larger than the previous size, the step continues to wait and checks the size again after the polling interval has elapsed.
- If the file is the same size as it was on the previous check, the step notes that the file size has not changed; the size has been the same two times.

The step looks at the value of the **File size verification count**. This value is the number of times that the file size must remain the same before the step can consider the file fully copied. If the file size verification count is equal to the number of times the file size has been the same, the step considers the file complete.

When the file is complete, the step renames the file based on the value of the **Retrieved file** property and copies it back into the spool directory for the job. Then the step completes its processing. It might pass the job to another step or terminate a branch of the workflow.

↓ Note

RICOH ProcessDirector offers separately priced features that provide interfaces with specific external programs. These include Avanti Slingshot Connect, Quadient Inspire Connect, PitStop Connect, and Ultimate Impostrip® Connect. Postal Enablement works with more than one external program.

- Avanti Slingshot Connect provides an interface with the Avanti Slingshot Management Information System.
- PitStop Connect provides an interface with Enfocus PitStop Server 10 or higher.
- Postal Enablement provides an interface with postal cleansing software such as Bell and Howell BCC Mail Manager and TEC Mailing Solutions MailPreparer.
- Quadient Inspire Connect provides an interface with Quadient Inspire Designer, version 8 and higher.
- Ultimate Impostrip® Connect provides an interface with Ultimate Impostrip® Automation or Scalable on Windows.

RICOH ProcessDirector does not supply the external program software with these features.

Setting up external programs

Before you start the configuration tasks for an external step, you need to set up the external program that the external step calls.

Install the external program and, if applicable, the application/secondary server software before you start this procedure.

Important points to remember when you install the base product or application/secondary servers and the other application:

- When you install the RICOH ProcessDirector software on a Linux or AIX system, the installation program creates the RICOH ProcessDirector system user (**aiw1** is the default). When you verify the operation of an external program that runs on a Linux or AIX system, use the RICOH ProcessDirector system user to do your verification testing.

- RICOH ProcessDirector does not automatically create a system user account for RICOH ProcessDirector on a Windows application server. The installation program prompts for the name of the user account to use as the RICOH ProcessDirector user. Verify the operation of the external program from the user account for the RICOH ProcessDirector user on the Windows application server.
- When the RICOH ProcessDirector primary computer is a Linux or AIX system, make sure that the RICOH ProcessDirector system user and the user that the other application uses have the appropriate permissions for the directories and file systems that they use.
- If the RICOH ProcessDirector primary computer is a Windows system and the other application is on a mapped network drive, edit the `mountDrives.bat` file to map the network drive whenever RICOH ProcessDirector starts. The file is located in the `c:\aiw\aiw1\bin\` directory.
- Make sure that the RICOH ProcessDirector system user on a Linux or AIX primary computer or RICOH ProcessDirector on a Windows primary computer can find and execute the other application. Use one of these methods:
 - Include the full path to the application in the command that you create.
 - Add the application to the PATH environment variable for the Windows primary computer or the RICOH ProcessDirector system user on a Linux or AIX primary computer.
This option works best if the other application runs on the same system as the primary server. If the application runs on an application/secondary server, this option only works if you mount the directory that the other application is installed in.
 - On a Linux or AIX primary computer, create a symbolic link to the other application, storing the link in `/aiw/aiw1/bin`.
Storing a symbolic link in that directory provides some important benefits. RICOH ProcessDirector adds that directory to the PATH environment variable when it creates the RICOH ProcessDirector system user, so the application can be recognized without separately editing the PATH. The directory is not deleted when you apply service, although some of the files in it might be updated. The link file should not be affected.

In addition, application and secondary servers can access this directory. As a result, they can also use the symbolic link to locate the external application if it is located within the `/aiw` file system. The external application cannot run on both a Linux or AIX primary computer and a Windows application server.

To set up the external program:

1. Use the documentation for the external program and verify that it runs without errors as a standalone program.
2. If you plan to use **RunExternalProgram**:
 1. If the application runs on a Linux or AIX computer, log in as the RICOH ProcessDirector system user. If it runs on a Windows computer, log in as an administrator.
 2. Compose the command that you can use to invoke the program from the command line and use it to submit an input file to the program.
Record the command that works correctly so you can use it as the model when you configure your step template.
3. **Optional:** Create a control file template for the step to use.

If the external program reads the property values that it requires from a parameter file that accompanies the print file, the **RunExternalProgram** step can build that file. To generate the file, the step uses a control file template. To create the control file template:

- 1) Make a copy of a parameter file that the application can use.
- 2) Replace all of the property values in the file with the RICOH ProcessDirector symbol notation representation of the property. If the value is the file name for a file in the job, replace it with the appropriate RICOH ProcessDirector method call to determine the file name.
During processing, the step uses the template to build the parameter file by resolving all the references and filling in the values for the current job.

- 3) Copy the control file template into a directory in the RICOH ProcessDirector shared file system (/aiw/aiw1/ (AIX and Linux) or C:\aiw\aiw1\ (Windows).)

Sample control file templates for external programs are installed in /aiw/aiw1/samples/external_programs/ (AIX and Linux) or C:\aiw\aiw1\samples\external_programs\ (Windows). You can copy these files to the /aiw/aiw1/control_files/external_programs/ (AIX and Linux) or C:\aiw\aiw1\control_files\external_programs\ (Windows) directory and customize them, or add your own control files to the /aiw/aiw1/control_files/external_programs/ (AIX and Linux) or C:\aiw\aiw1\control_files\external_programs\ (Windows) directory. Note the directory location of your control file template.

- 4) On a Linux or AIX system, verify that the file permissions for the control file template provide access for the other class of system users. For example, set the permissions for the control file template to **666**.
 - 5) Make sure one external command passes the control file in the parameter it expects (the -o option for including additional job properties, for example).
4. **Optional:** On a Linux or AIX system, create a symbolic link to the application on the primary computer:

- 1) Log on to the system that the RICOH ProcessDirector base product runs on as the RICOH ProcessDirector system user.
- 2) Use the **stopaiw** command to stop the primary server.
- 3) Use this command to create the symbolic link:

```
ln -s application_file /aiw/aiw1/bin/link_file
```

Replace *application_file* with the full path to the executable file for the application, and replace *link_file* with the name of the symbolic link file you want to create. The executable file must be located within the /aiw file system.

- 4) Use the **startaiw** command to start the primary server.

3. If you plan to use **RunHotFolderApplication**:

1. Log on to the primary computer. On AIX or Linux systems, log in as the RICOH ProcessDirector system user.
2. Copy or transfer a sample print file into the input folder for the application.
3. Check the other application to make sure that it starts to process the file.
4. Monitor the output folder for the resulting file. When it arrives, copy or transfer it to another directory, then verify that it is correct.

5. Navigate to the directory that holds the log files for the other application and make sure that you can open and view the log files.

External steps

External commands

If you choose to use the **RunExternalProgram** step template, figure out the command that you want to use before you start to create your step template. Here are some example commands that use RICOH ProcessDirector symbol notation.

3

Copy a newer print file to a destination

In this example, the Linux **cp** command only copies the *JobNumber.print.pdf* file from the spool directory for the job when the source file is newer than the destination file:

```
cp -u ${getAbsoluteFileName(print, pdf, read)} /tmp/jobarchives
```

`${getAbsoluteFileName(print, pdf, read)}` is the RICOH ProcessDirector symbol formula that returns the name of the PDF print file in the spool directory. If the PDF print file does not exist in the spool directory when the external program runs, an error occurs.

Update a print file and write it to the spool directory

In this example, the external program **myprogram** reads the *JobNumber.print.pdf* file from the spool directory and writes an updated version of the file to the spool directory using redirection:

```
myprogram -i ${getFileName(print, pdf, read)} >
${getFileName(print, pdf, write)}
```

`${getFileName(print, pdf, read)}` is the RICOH ProcessDirector symbol formula that returns the name of the PDF print file in the spool directory. If the PDF print file does not exist in the spool directory when the external program runs, RICOH ProcessDirector returns the name of the input file for the job, which is *JobNumber.print.unknown*. If that file does not exist in the spool directory, an error occurs.

Read the overrides file for a print job and write a statistics file to the spool directory

In this example, the external program **auditstatistics** reads the *JobNumber.overrides.text* file from the spool directory and writes a new statistics file *JobNumber.statistics.text* to the spool directory:

```
auditstatistics inputfile=${getFileName(overrides, text, read)}
outputfile=${getFileName(statistics, text, write)}
```

Use a separate file to pass property values to an external program with the print file

In this example, the Linux **cp** command produces a CSV file for which a corresponding RICOH ProcessDirector control file exists. The control file lists the symbols for nine RICOH ProcessDirector job properties:

```
cp ${getControlFileName()} /aiw/aiw1/samples/${Job.ID}.info.csv
```

`${getControlFileName()}` is the RICOH ProcessDirector symbol formula that returns the name of the resolved control file. RICOH ProcessDirector generates the control file from the control file template that you specify.

The control file can use the **getChildFileName** method to return the name of a child file so that the external program can write a file to the **children** subdirectory in the spool directory. The external program must write the file names of child-job files in this format:

```
JobNumber.UsageType.DataType.n,Job.JobType=JobTypeName
```

JobTypeName is the name of the workflow that the child job requires. The workflow must exist and it must be enabled.

When you add a step based on the **RunExternalProgram** step template to a workflow using the **Job Types** property notebook, the properties that show **[Receive]** in the job defaults change to reflect the actual phase to which you add the step.

Invoke a command using a full path name

This example uses a full path name to invoke the user-supplied external program **mytransform** to transform a file to PDF format. **mytransform** reads the names of the input and output files from a control file:

```
/opt/myprograms/bin/mytransform -c ${getControlFileName()}
```

Archive a print file to a Windows system

In this example, the external step uses the Windows **copy** command to write a copy of the *JobNumber.print.pdf* file from the spool directory to an archive directory on the Windows system:

```
copy ${getAbsoluteFileName(print,pdf,read)} d:\archive\pdf
```

Note

- The Windows system directory must have RICOH ProcessDirector installed and the Windows application server must be connected to the primary server.
- Specify the Windows directory by using the native Windows format for the directory name.
- Use native Windows commands instead of Windows Services for UNIX (SFU) commands where appropriate; for example, use the **copy** command instead of the **cp** command. Not all SFU commands are available on Windows systems.
- Depending on the command, a directory on the Windows system might not need to exist. The command can create the directory on the Windows system.
- Make sure that you tune the step template to only run on a Windows application server. Because of the different formats for directory names on Linux and AIX, a step based on this step template will fail if RICOH ProcessDirector tries to run the step on a non-Windows system.

Using RICOH ProcessDirector methods

When RICOH ProcessDirector creates a job, it also creates a spool directory. The spool directory contains a copy of the input file and other files that provide information about the job. Steps and commands can access the various files in the spool directory and use them during processing. They can also update existing spool files and write new files to the directory.

RICOH ProcessDirector provides these methods for reading and writing files in the spool directory for a job:

- **getFileName**
- **getAbsoluteFileName**
- **getCurrentFile**
- **getCurrentAFPFile**
- **getControlFileName**

- `getChildFileName`
- `getChildFileStem`

Spool directory names

The default spool directory that RICOH ProcessDirector creates for a job is `/aiw/aiw1/spool/default/JobNumber` (AIX and Linux) or `C:\aiw\aiw1\spool\default\JobNumber` (Windows).

For example:

- `/aiw/aiw1/spool/default/10000006` (AIX or Linux)
- `C:\aiw\aiw1\spool\default\10000006` (Windows)

The default spool directory that it creates for a child job is `/aiw/aiw1/spool/default/JobNumber.Number` (AIX and Linux) or `C:\aiw\aiw1\spool\default\JobNumber.Number` (Windows). For example:

- `/aiw/aiw1/spool/default/10000006.4` (AIX or Linux)
- `C:\aiw\aiw1\spool\default\10000006.4` (Windows)

RICOH ProcessDirector messages and the properties notebook for a job refer to the spool directory as the **Root file path**.

RICOH ProcessDirector always uses these naming conventions for the names of the default spool directories. Authorized users cannot change the default naming conventions.

You can create additional spool directories. Additional spool directories have names in this format: `/aiw/aiw1/spool/SpoolName/JobNumber` (AIX and Linux) or `C:\aiw\aiw1\spool\SpoolName\JobNumber` (Windows) and `/aiw/aiw1/spool/SpoolName/JobNumber.Number` (AIX and Linux) or `C:\aiw\aiw1\spool\SpoolName\JobNumber.Number` (Windows). *SpoolName* is the subdirectory name that the authorized user assigned.

Spool file names

When RICOH ProcessDirector processes a job, it generates spool files for the job. Formats for RICOH ProcessDirector spool file names are:

Syntax 1, parent or standalone job:

```
JobNumber.Usagetype.Datatype
```

Syntax 2, child job:

```
JobNumber.Usagetype.Datatype.ChildGroupID
```

where:

Usagetype

Describes the purpose or role of the spool file within the job. Usage-type keywords are case-sensitive. Although RICOH ProcessDirector has usage-type keywords other than the ones listed here, these are the usage-type keywords for spool files that external programs might want to read from or write to:

aiwlist

Contains an entry for each input file in the job and its submit type. Depending on the job, RICOH ProcessDirector might have generated one or more of the input files or they might have accompanied the input file that contained the initial job data. The **aiwlist** file can be useful in problem determination.

This example shows what the file contents might look like when RICOH ProcessDirector generates one or more of the input files. The information in the first column is the submit type.

AIX and Linux:

```
Data;/aiw/aiw1/System/hf/PDF/Staged/
sample.PRD.AIW_TS.2006-04-27-08.57.31.476
Other;/aiw/aiw1/System/hf/PDF/Staged/
sample.JCL.AIW_TS.2006-04-27-08.57.31.593
List;/aiw/aiw1/System/hf/PDF/Staged/
sample.lst.AIW_TS.2006-04-27-08.57.31.534
AIWList;/aiw/aiw1/System/hf/PDF/Staged/
AIWList.AIW_TS.2006-04-27-08.57.31.794
```

Windows:

```
Data;C:\aiw\aiw1\System\hf\PDF\Staged\
sample.PRD.AIW_TS.2006-04-27-08.57.31.476
Other;C:\aiw\aiw1\System\hf\PDF\Staged\
sample.JCL.AIW_TS.2006-04-27-08.57.31.593
List;C:\aiw\aiw1\System\hf\PDF\Staged\
sample.lst.AIW_TS.2006-04-27-08.57.31.534
AIWList;C:\aiw\aiw1\System\hf\PDF\Staged\
AIWList.AIW_TS.2006-04-27-08.57.31.794
```

control

A special RICOH ProcessDirector-generated control file. It contains processing parameters that were specified in a control file template that accompanied the input file for the job. RICOH ProcessDirector generates the control file from the file specified in the **External control file template** property for the job. When it creates the control file, it resolves any RICOH ProcessDirector-specific entries from the control file template. For example, the PDF workflow runs a RunExternalProgram step during the Prepare phase. The RunExternalProgram step specifies this **External command** property:

```
/aiw/aiw1/bin/copy_files.pl ${getControlFileName()} /aiw/aiw1/samples/
${Job.ID}.info.csv
```

The RunExternalProgram step specifies this **External control file template** property:

```
/aiw/aiw1/control_files/external_programs/job_info.cfg (Linux)
C:\aiw\aiw1\control_files\external_programs\job_info.cfg (Windows)
```

The **job_info.cfg** control file template contains this information:

```
Job.ID,Job.Name,Job.JobType,Job.SubmitTime,Job.RequestedPrinter,
Job.InputFile.Size,Job.JobSize,Job.TotalPages,Job.TotalSheets

${Job.ID},${Job.Name},${Job.JobType},${Job.SubmitTime},
${Job.RequestedPrinter},${Job.InputFile.Size},${Job.JobSize},
${Job.TotalPages},${Job.TotalSheets}
```

When RICOH ProcessDirector creates a control file from this template, it resolves the values for the `${Job.ID}`, `${Job.Name}`, `${Job.JobType}`, `${Job.SubmitTime}`, `${Job.RequestedPrinter}`, `${Job.InputFile.Size}`, `${Job.JobSize}`, `${Job.TotalPages}`, and `${Job.TotalSheets}` symbols to the actual RICOH ProcessDirector job properties. For example:

```

${Job.ID}=10000001
${Job.Name}=Demo.pdf
${Job.JobType}=PDF
${Job.SubmitTime}=10:02.35
${Job.RequestedPrinter}=Sample
${Job.InputFile.Size}=2608549
${Job.JobSize}=26
${Job.TotalPages}=26
${Job.TotalSheets}=26

```

Note

- Not all jobs that RICOH ProcessDirector creates have associated control files in their spool directories.

overrides

Contains a copy of the job-specific parameters that accompany the input file for the job. RICOH ProcessDirector can use these parameters with a control file to set job properties for the job.

For example, jobs that RICOH ProcessDirector receives through Download for z/OS or AFP Download Plus when the AFP Support feature is installed include *datasetname*.JCL files that the Download daemon generates. These files contain information that represents the values that were specified for JCL parameters. This example shows the contents of an overrides file that contains JCL values for parameters:

```

"-odatat=line -ofileformat=record -occ=yes -occtype=a -ochars=GF10
-opagedef=P1A 06462 -of=F1A10110 -ocop=001 -odatac=block -ojobn=PAYROLL
-ous=ADMIN1 -ono=BLD PDEVL -opr=KYSMITH -opa=forms=STD,class=C,
destination=CHI3,jobid=JOB 02357,OUTGRP=NEXT"

```

A control file, such as `receive_jcl_jobtype.cfg` in `/aiw/aiw1/samples/rules/` (AIX and Linux) or `C:\aiw\aiw1\samples\rules\` (Windows) that RICOH ProcessDirector provides, can process the overrides file.

Note

- Not all jobs that RICOH ProcessDirector creates have associated overrides files in their spool directories.

print

A print file that contains the data for the job. The data might match the format of the input file that RICOH ProcessDirector received or it might be in a different format. The format depends on the processing that has occurred for the job. For example, an input file might be converted to PostScript during processing.

In addition to the usage-type keywords that RICOH ProcessDirector uses, authorized users can specify their own usage-type keywords when they use RICOH ProcessDirector methods to manipulate files in the spool directory. However, user-defined keywords should not be variations of the RICOH ProcessDirector keywords if RICOH ProcessDirector will further process a file that an external program writes to the spool directory. For example, do not use a usage type of **PRINT** or **Print**; RICOH ProcessDirector only recognizes **print**.

Datatype

Describes the content or the data stream type of the spool file. Whether the spool directory for a given job contains a spool file of a specific data type depends on the processing that has occurred

for the job. Datatype keywords are case-sensitive. The keywords that RICOH ProcessDirector provides are:

afp

Advanced Function Presentation (AFP) format. The spool file contains job print data.

gif

Graphic Interchange Format (GIF). The spool file contains image data.

jdf

Job Definition Format (JDF). The spool file contains job-specific parameters from the job ticket.

jpeg

Joint Photographic Experts Group (JPEG) format. The spool file contains image data.

json

JavaScript Object Notation (JSON) format.

linedata

Line-data format or mixed-mode format. The spool file contains job input data.

mjm

Multipurpose Internet Mail Extensions (MIME) package with data in Portable Document Format (PDF), Job Definition Format (JDF), and Job Messaging Format (JMF). The spool file contains job input data and job-specific parameters from the job ticket.

pcl

Printer control language (PCL) format. The spool file contains job input data.

pdf

Portable Document Format (PDF). The spool file contains job input data.

ps

PostScript format. The spool file contains job input data.

text

The data stream is a plain text file. The spool file can contain non-job data, such as processing parameters or information created by an external program.

tiff

Tagged Image File Format (TIFF). The spool file contains image data.

xml

Extended Markup Language (XML) format. The spool file contains well-structured XML.

zip

Data-compression and archiving format for one or more files. The spool file can contain job input data, image data, or non-job data, depending on the format of the files in the zip file.

unknown

The data stream is of an unknown format. This is the datatype keyword that RICOH ProcessDirector initially sets for the spool file that is a copy of the original input file for the job. For more detailed information, see the description of the **Input data stream** job property. Files with a datatype keyword of **unknown** might only be in the spool directory for a very short period of time after RICOH ProcessDirector creates the job.

In addition to the datatype keywords that RICOH ProcessDirector uses, authorized users can specify their own datatype keywords when they use RICOH ProcessDirector methods to let external programs write files to the spool directory. However, user-defined keywords for the data type should not be variations of the RICOH ProcessDirector keywords if RICOH ProcessDirector will further process a file that an external program writes to the spool directory. For example, do not use a datatype of **PDF** or **Pdf**; RICOH ProcessDirector only recognizes **pdf**.

3

ChildGroupID

A unique numeric value that RICOH ProcessDirector uses to identify and group all files that relate to a child job that an external program creates. The child group ID keyword is in addition to the datatype keyword; parent jobs or single jobs do not include a child group ID keyword.

getFileName and getAbsoluteFileName syntax

You can use the **getFileName** and **getAbsoluteFileName** methods to return the name of a specific file in the spool directory. This is useful because RICOH ProcessDirector assigns a unique job number for each job and includes the job number in file names. The main difference between the two methods is that **getAbsoluteFileName** fails if the specifically requested file does not exist; **getFileName** can return the name of an alternate spool file.

To use the RICOH ProcessDirector **getFileName** method or the **getAbsoluteFileName** method, use symbol notation to refer to them as RICOH ProcessDirector symbol formulas. Use one of these syntaxes:

Syntax 1, single search:

```
#{getFileName(UsageKeyword, DatatypeKeyword, FileaccessKeyword)}
#{getAbsoluteFileName(UsageKeyword, DatatypeKeyword, FileaccessKeyword)}
```

Syntax 2, iterative search:

```
#{getFileName(UsageKeyword, (DatatypeKeyword1, DatatypeKeyword2...),
FileaccessKeyword)}
#{getAbsoluteFileName(UsageKeyword, (DatatypeKeyword1,
DatatypeKeyword2...), FileaccessKeyword)}
```

where:

UsageKeyword

A case-sensitive keyword that identifies the usage type of the spool file. The values that authorized users can specify are the same values that the usage-type portion of the spool file name can contain. For example, you can specify these RICOH ProcessDirector-supplied keywords: **aiwlist**, **control**, **overrides**, or **print**. You can also specify your own user-defined keywords.

DatatypeKeyword or (*DatatypeKeyword1, DatatypeKeyword2...*)

A case-sensitive keyword that specifies the datatype of the spool file. The values that authorized users can specify are the same values that the datatype portion of the spool file can contain. For

example, you can specify RICOH ProcessDirector-supplied keywords, such as **jdf**, **pdf**, or **gif**. You can also specify your own user-defined keywords.

If you want RICOH ProcessDirector to continue to search for another spool file if it does not find a spool file of the initial type, you can specify multiple datatype keywords. Separate them with commas and enclose them in parentheses. For example:

```
${getFileName(UsageKeyword, (print, pdf), FileAccessKeyword)}
```

In this example, the **getFileName** method first looks for a spool file with a datatype of **print**. If it does not find a matching spool file name, it looks for a spool file with a datatype of **pdf**. If it still does not find a matching spool file name, it looks for a spool file with a datatype of **unknown**. If that spool file does not exist, RICOH ProcessDirector issues an error message.

↓ Note

The processing point at which RICOH ProcessDirector attempts to resolve the name of the file determines whether the method locates the file. If the workflow includes the **SetJobPropsFromTextFile** step, use the **getAbsoluteFileName** method instead of the **getFileName** method. The **SetJobPropsFromTextFile** step tries to resolve the file name that the **getFileName** method represents immediately after the input device creates the job, and not all spool files are available at that point in processing. RICOH ProcessDirector does *not* try to resolve the file name that the **getAbsoluteFileName** method represents immediately after job creation.

You can also specify multiple datatype values with the **getAbsoluteFileName** method, such as:

```
${getAbsoluteFileName(UsageKeyword, (pdf, postscript, text), FileAccessKeyword)}
```

The **getAbsoluteFileName** method looks for spool files in the same manner as the **getFileName** method, with one exception. If no spool files exist with any of the specified datatypes, the default is that the **getAbsoluteFileName** method does not look for a spool file with a datatype of **unknown**.

FileAccessKeyword

The file-access keyword, which is either **read** or **write**, specifies whether the external program reads the file from the spool directory or writes the file to the spool directory. RICOH ProcessDirector creates files in subdirectories of the spool directory, depending on whether the external program reads or writes a file:

checkpoints subdirectory

When the external program makes changes to the file through a write operation, RICOH ProcessDirector moves the original version of the file to this subdirectory. If an authorized user subsequently requests a process-again action, RICOH ProcessDirector can restore the original spool file from the **checkpoints** subdirectory. This makes sure that the external program has the same input available for its write operation as the first time that it processed the file.

tmp subdirectory

When the external program creates a new file, RICOH ProcessDirector stores the new file in this subdirectory until the external step that calls the external program completes. When the step completes, RICOH ProcessDirector moves the new file to the spool directory.

 **Note**

External programs that create child jobs must write them to the `children` subdirectory of the spool directory for the job. The file name of the child job that the external program creates must be in this format:

```
JobNumber.Usagetype.Datatype,ChildGroupID,Job.JobType=JobType
```

The *ChildGroupID* must be the same for all files associated with a single child job. If the external program generates more than one child job, it must increment the *ChildGroupID* for each set of files associated with a new child job.

The external program appends **,Job.JobType=** to the file name, which is the database name for the workflow property. **JobType** value is the name of the workflow that the child job requires; this workflow must exist and it must be enabled.

getFileName and getAbsoluteFileName read examples

All of these examples use the spool directory:

- `/aiw/aiw1/spool/default/10000006` (AIX and Linux)
- `C:\aiw\aiw1\spool\default\10000006` (Windows)

The spool directory contains these spool files:

```
10000006.banner_attributes.text
10000006.control.text
10000006.overrides.text
10000006.print.pdf
10000006.print.unknown
10000006.resources.pdf
10000006.resources.log
10000006.resources.outlines
```

For the purpose of these examples, the value of the **Input data stream** job property was not set through any manner, such as through a job default in the workflow or a detection step. Therefore, a spool file with a datatype of **unknown** is present in the spool directory:

- To return the path and name of the control file for the job, which is `/aiw/aiw1/spool/default/10000006/10000006.control.text` (AIX and Linux) or `C:\aiw\aiw1\spool\default\10000006\10000006.control.text` (Windows), so that the external program can do a read operation on the file:

```
`${getFileName(control, text, read)}`
`${getAbsoluteFileName(control, text, read)}`
```

- To return the path and name of the input file for the job, which is `/aiw/aiw1/spool/10000006/10000006.print.unknown` (AIX and Linux) or `C:\aiw\aiw1\spool\10000006\10000006.print.unknown` (Windows), so that the external program can do a read operation on the file:

```
`${getFileName(print, unknown, read)}`
`${getAbsoluteFileName(print, unknown, read)}`
```

This examples use the spool directory:

- `/aiw/aiw1/spool/default/10000009` (AIX and Linux)
- `C:\aiw\aiw1\spool\default\10000009` (Windows)

The spool directory contains these spool files:

```
10000009.banner_attributes.text
10000009.control.text
10000009.overrides.text
10000009.print.ps
10000009.resources.pdf
10000009.resources.log
10000009.resources.outlines
```

In this example, the type of the input data stream was detected and `10000009.print.ps` is the copy of the original input file. To search for an input file in one of the supported PostScript formats:

```
#{getFileName(print, (pdf, ps), read)}
#{getAbsoluteFileName(print, (pdf, ps), read)}
```

Both methods first look for a spool file with the name `10000009.print.pdf`. Because that spool file is not present, the methods continue to look for a spool file with the name `10000009.print.ps`.

getFileName and getAbsoluteFileName write examples

All of these examples use the spool directory:

- `/aiw/aiw1/spool/default/10000006` (AIX and Linux)
- `C:\aiw\aiw1\spool\default\10000006` (Windows)
- To set up for a write operation by an external program that writes a new statistics record file to the spool directory:

```
#{getFileName(statistics, record, write)}
#{getAbsoluteFileName(statistics, record, write)}
```

RICOH ProcessDirector provides the external program with the path and file name `/aiw/aiw1/spool/default/10000006/10000006.statistics.record` on AIX or Linux and `C:\aiw\aiw1\spool\default\10000006\10000006.statistics.record` on Windows.

- To set up for a write operation by an external program that updates the PDF print file in the spool directory:

```
#{getFileName(print, pdf, write)}
#{getAbsoluteFileName(print, pdf, write)}
```

RICOH ProcessDirector checkpoints a file with the name `10000006.print.pdf`, if it exists, into the checkpoints subdirectory of the spool directory for the job.

RICOH ProcessDirector also provides the external program with the path and file name `/aiw/aiw1/spool/default/10000006/tmp/10000006.print.pdf` on AIX or Linux and `C:\aiw\aiw1\spool\default\10000006\tmp\10000006.print.pdf` on Windows.

- If the program fails, the spool directory for the job is unchanged because the external program wrote to a file that RICOH ProcessDirector moved to the `tmp` subdirectory. RICOH ProcessDirector only moves the file to the spool directory if the external step that calls the external program completes successfully.
- If the program succeeds, RICOH ProcessDirector moves the file from the `tmp` subdirectory into the spool directory. If a file by that name already exists in the spool directory, RICOH ProcessDirector moves that file to the checkpoint subdirectory, then moves the newly created file from `tmp` to the spool directory for the job.

getCurrentFile syntax

The **getCurrentFile** method returns the name of a print file in the spool directory for the job. If a file exists that contains a page range selected from the original print file, it returns the name of that file. If not, it returns the name of the original print file. This is useful for allowing the same command to work on the full job as well as a subset of the job, as is often the case with reprints.

Authorized users **can** use this format for the method:

```
${getCurrentFile(datatype)}
```

where:

datatype

The datastream of the print file. AFP can be specified, which makes this method identical to the **getCurrentAFPFile** method. You can use symbolic notation, such as `${Job.InputDataStream}`, for this parameter.

getCurrentFile example

To submit a job to a Passthrough printer using **lpr** on a UNIX-based system, you could use this command for the value of the **Printer command** property:

```
lpr -Pmyprinter ${getCurrentFile(${Job.InputDataStream})}
```

The first time the job is sent to the printer, the entire job prints. If the job is processed again to print a subset of pages, only the subset is printed.

getCurrentAFPFile syntax

The **getCurrentAFPFile** method returns the name of an AFP print file in the spool directory for the job. If a file that contains a page range selected from the original print file exists, it returns the name of that file. If not, it returns the name of the original print file. This is useful for working with reprinted jobs, which can contain a subset of the original job.

The method uses no parameters. Authorized users must always use this format for the method:

```
${getCurrentAFPFile()}
```

getCurrentAFPFile example

A workflow contains an external step in the Print phase, before the PrintJobs step, that specifies this property and value:

External command [Print][RunExternalProgram]

Value: `itm_driver -C ${getControlFileName()} -F"-itm_in_files ${getCurrentAFPFile()} -itm_out_files ${getFileName(print, pdf, write)}"`

The first time that a job of this type is printed, RICOH ProcessDirector copies the print file to the tmp subdirectory as `10000004.print.afp`. The external command converts the AFP file to PDF for printing.

When the job is reprinted, the operator selects a range of pages from the original print file. RICOH ProcessDirector copies only the selected pages to the tmp subdirectory as `10000004.print_range.afp`. The external command converts `10000004.print_range.afp` to PDF format.

getControlFileName syntax

The **getControlFileName** method returns the name of the resolved control file for the job.

The method uses no parameters. Always use this format for the method:

```
${getControlFileName() }
```

getControlFileName example

This example uses this spool directory:

- /aiw/aiw1/spool/default/10000003 (AIX or Linux)
- C:\aiw\aiw1\spool\default\10000003 (Windows)

It describes how RICOH ProcessDirector creates files and generates values when you use the **getControlFileName** method.

A workflow contains an external step in the Prepare phase that specifies these properties and values:

External control file template [Prepare][RunExternalProgram]

Value:

- /aiw/aiw1/control_files/external_programs/job_info.cfg (AIX or Linux)
- C:\aiw\aiw1\control_files\external_programs\job_info.cfg (Windows)

External command [Prepare][RunExternalProgram]

Value:

- cp \${getControlFileName()} /aiw/aiw1/samples/\${Job.ID}.info.csv (AIX or Linux)
- copy \${getControlFileName()} C:\aiw\aiw1\samples\\${Job.ID}.info.csv (Windows)

Just before running the external step, RICOH ProcessDirector copies the external control file template to the tmp subdirectory of the spool directory and resolves any symbols that the control file template contains. This is the procedure that RICOH ProcessDirector uses to create the resulting control file. For example, it generates this file: For example, it generates this file on AIX or Linux:

```
/aiw/aiw1/spool/default/10000003/tmp/job_info.control.text
```

And this file on Windows:

```
C:\aiw\aiw1\spool\default\10000003\tmp\job_info.control.text
```

When RICOH ProcessDirector creates the job that uses the workflow and sets its initial property values, it uses the value that the **getControlFileName** method returned to resolve the external command. On AIX or Linux, the command is:

```
cp /aiw/aiw1/spool/default/10000003/tmp/job_info.control.text /aiw/aiw1/samples/10000003.info.csv
```

Or, on Windows, to this:

```
copy C:\aiw\aiw1\spool\default\10000003\tmp\job_info.control.text C:\aiw\aiw1\samples\10000003.info.csv
```

getChildFileName syntax

You can use the **getChildFileName** method when an external program creates only one child job and its associated files. This method returns a file name for a child job. When the external program starts to write the data for the child job, it writes the data to a file with the name that the **getChildFileName** method returns.

To use the RICOH ProcessDirector **getChildFileName** method, use symbol notation to reference it as a RICOH ProcessDirector symbol formula. Use this syntax:

```
#{getChildFileName(UsageKeyword, DatatypeKeyword, ChildGroupID)}
```

where:

UsageKeyword

A case-sensitive keyword that identifies the usage type of the spool file. The use of *UsageKeyword* with the **getChildFileName** method is the same as described earlier in this topic for spool files and the **getFileName** and **getAbsoluteFileName** methods.

DatatypeKeyword

A case-sensitive keyword that identifies the datatype of the spool file. The use of *DatatypeKeyword* with the **getChildFileName** method is the same as described earlier in this topic for spool files and the **getFileName** and **getAbsoluteFileName** methods.

ChildGroupID

A unique numeric value that RICOH ProcessDirector uses to identify the files that belong to the same child job. Typically, when the external program generates a single child job, the value is set to **1**.

getChildFileStem syntax

You can use the **getChildFileStem** method when the external program creates more than one child job or when you do not know how many child jobs the external program will generate. This method generates a root child job file name that an external program can use to generate as many child job file names as it needs.

To use the RICOH ProcessDirector **getChildFileStem** method, use symbol notation to reference it as a RICOH ProcessDirector symbol formula. Use this syntax:

```
#{getChildFileStem(UsageKeyword, DatatypeKeyword)}
```

where:

UsageKeyword

A case-sensitive keyword that identifies the usage type of the spool file. The use of *UsageKeyword* with the **getChildFileStem** method is the same as described earlier in this topic for spool files and the **getFileName** and **getAbsoluteFileName** methods.

DatatypeKeyword

A case-sensitive keyword that identifies the datatype of the spool file. The use of *DatatypeKeyword* with the **getChildFileStem** method is the same as described earlier in this topic for spool files and the **getFileName** and **getAbsoluteFileName** methods.

Setting up step templates for external steps

You create a step template that contains the command that calls the external program. Then, you tune the step template so that it will run on the Linux, AIX, or Windows system on which the external program is installed.

Setting up step templates for external steps that use the command line or control files

Use this process to set up a step template for external steps that use the command line or a control file to pass parameters between RICOH ProcessDirector and the external program.

Note

- If each job has different requirements for external programs, you do not need to set up a step template. Use the **RunExternalProgram** step template to add a step to a workflow and then update the **External command**, **Valid return codes**, and **External program language** properties for the step, as described below.

To set up a step template for an external step:

1. Click the **Workflow** tab.
2. In the left pane, click **Step Templates**.
3. Select the checkbox next to the **RunExternalProgram** step template.
4. Click **Copy**.
5. Specify a name and description for the new step template.
6. Click the **External** tab.
7. Update the **External command** property.

This value is the actual command and any command-line parameters that RICOH ProcessDirector issues to run the external program. The command string can include RICOH ProcessDirector symbol notation.

8. If you created a control file template for use with the external program, update the **External control file template** property. Set the value to the directory location and name of the control file template. If the external program only receives its parameters as command line arguments, delete any value for this property.
9. Update the **Valid return codes** property.

In this context, a valid return code is any return code from the external program that does not require a user action. Separate multiple return code numbers with commas. RICOH ProcessDirector moves the job to the error state if the external program returns any values that are not included in this value. It also issues a message in the log for the job to alert you to a problem that the external program reported.

For example, if the value of the valid-return-codes property is **0,4**, and the external program finishes with a return code of **16**, RICOH ProcessDirector issues a message similar to this:

```
AIWI6073E External step cp /aiw/aiw1/spool/default/10000016/10000016.print.
unknown /archive/directory finished with a return code of 16, which is a defined as
an error.
```

10. If the installation has specific language requirements, you can also instruct the external program to return messages in a language that it supports. The drop-down list for the **External program language** property lists the languages that RICOH ProcessDirector supports.
11. Click **OK**.
12. Select the new step template and click **Enable**.

Setting up step templates for external steps that use hot folders

3

Use this process to set up a step template for external steps that use hot folders to pass input and output between RICOH ProcessDirector and the external program.

To set up a step template for an external step:

1. Click the **Workflow** tab.
2. In the left pane, click **Step templates**.
3. Right-click the **RunHotFolderApplication** step template and select **Copy**.
4. Specify a name and description for the new step template.
5. Click **Hot Folder**.
6. Update the **Sending folder** property.

The value of this property is the name of the input hot folder for the external program. RICOH ProcessDirector places the job in this folder to submit it to the external program.

↓ Note

- The value must be the name of a folder that exists. RICOH ProcessDirector does not create the folder.
- The folder must be accessible to both RICOH ProcessDirector and the external program. It can be:
 - In the RICOH ProcessDirector shared file system, /aiw/aiw1/ (AIX and Linux) or C:\aiw\aiw1\ (Windows), on the computer that has the primary server installed.
 - In a file system that is shared using file sharing software, such as Samba.
- Even if the folder is on a Windows system, specify the path using AIX or Linux format. For example, if the folder is C:\Sending, type /Sending.
- If this value is null, no file is copied and the step waits.

7. Update the **File to send** property.

The value of this property is a symbol formula that resolves to the name of the file that RICOH ProcessDirector sends to the external program.

↓ Note

- The default value is `getAbsoluteFileName(print, pdf, read)`, which returns the name of the PDF print file in the spool directory. If the PDF print file does not exist in the spool directory when the external program runs, an error occurs.

8. Update the **Retrieval folder** property.

The value of this property is the name of the output hot folder for the external program. RICOH ProcessDirector retrieves the job from this folder after the external program has processed it.

Note

- The value must be the name of a folder that already exists. RICOH ProcessDirector does not create the folder.
- The folder must be accessible to both RICOH ProcessDirector and the external program. It can be:
 - In the RICOH ProcessDirector shared file system, `/aiw/aiw1/` (AIX and Linux) or `C:\aiw\aiw1\` (Windows), on the computer that has the primary server installed.
 - In a file system that is shared using file sharing software, such as Samba.
- Even if the folder is on a Windows system, specify the path using AIX or Linux format. For example, if the folder is `C:\Retrieval`, type `/Retrieval`.
- When you set the **Clean up retrieval folder** property to **No**, the file that the step retrieves can be in the retrieval folder when a job arrives at the step. The step retrieves the file and puts the file specified by the **File to send** property in the sending folder. The step then sends the job and the retrieved file to the next step in the workflow. The step does not wait for a new file to be placed in the retrieval folder.

9. Update the **Retrieval pattern** property.

The value of this property is the pattern-matching string that RICOH ProcessDirector uses to identify the output files to retrieve from the output hot folder of an external program.

For example, the value `${Job.ID}.*` matches any file whose file name is the same as the job ID, with any file extension.

10. Update the **Retrieved file** property.

The value of this property is a symbol formula that resolves to the name that RICOH ProcessDirector uses to rename the retrieved file.

11. Update the **Clean up retrieval folder** property.

When a job enters the step, this value tells RICOH ProcessDirector whether to remove any file in the retrieval folder whose file name matches the **Retrieval pattern**.

12. Update the **Application log file** property.

The value of this property is the folder where the external program writes log files. This value can be blank.

13. Update the **Poll interval** property.

The value of this property is the interval in seconds at which RICOH ProcessDirector checks the output hot folder for completed jobs.

14. Update the **File size verification count** property.

The value of this property is the number of times that RICOH ProcessDirector polls the output hot folder and finds a file that has not changed in size. RICOH ProcessDirector then decides that the file is complete.

15. Update the **Timeout interval** property.

The value of this property is the time limit in minutes for retrieving a job from the output hot folder of an external program. If the job has not been received or is not complete when the time limit is reached, the job is in error.

Note

- Make the timeout interval longer than the poll interval times the file verification count.
- If the timeout interval is 0, RICOH ProcessDirector waits forever.

16. Click **OK**.
17. Update any settings on the **Tuning** tab to run the step on the computers where the external program is installed.
See [Tuning step templates](#), p. 88 for instructions.
18. Select the new step template and click **Enable**.

Tuning step templates

Tuning a step template lets you specify how much system resource the step requires for processing. You can also specify which computers can run the steps created from the step template.

Step templates that let RICOH ProcessDirector access applications on other computers must be tuned to run on the server that those applications are installed on.

The **RunExternalProgram** and **RunHotFolderApplication** step templates (and any copies of them) can be tuned to run on the primary server, an application server, or a secondary server.

If the RICOH ProcessDirector primary computer is a Linux or AIX system, step templates installed by some features must be tuned to run on application servers. These step templates (and any copies of them) must be tuned to run on application servers:

- RunPitStopOnJob

Some step templates installed by features (including document processing features) can only run on the primary server. If your environment includes RICOH ProcessDirector application or secondary servers, you must tune these step templates to run only on the primary server:

- CreateInserterReprints
- CreateJobsFromDocuments
- CreateReprints
- GroupDocuments
- InsertJobs
- ReadBarcodeData
- ReadDocumentsFromDatabase
- ReadDocumentsFromParent
- SendInserterControlFile
- SetDocPropsFromConditions

- SetInsertProperties
- SetJobPropsFromOriginal
- UpdateDocumentsInDatabase
- WaitForDocumentCompletion
- WriteDocumentsToDatabase
- WriteInserterControlFile
- WritePropsToReportsDatabase

To tune a step template:

1. Click the **Workflow** tab.
2. In the left pane, click **Step Templates**.
3. Click the name of the step template that you want to tune.
4. Click the **Tuning** tab.
5. In the **Concurrent step limit** section, specify where the limits are set for the number of steps created from the step template that can run at the same time.
6. Click **OK**.

Setting up workflows for external steps

After you create a step template to call an external program and tune the step template to run on the server that the external program is installed on, you add a step that is based on the step template to a workflow. You then assign the workflow to an input device, or use another method to assign the workflow to specific jobs.

To set up a workflow for an external step:

1. Click the **Workflow** tab.
2. Determine whether you can use a copy of an existing workflow or if you need a new workflow. Do one of these:
 - Right-click one of the workflows and select **Copy**. Name the copy of the workflow, fill in or edit other values that you need, and click **Continue**.
 - Click **Add** and specify a name and description for the new workflow.
3. To add the external step:
 1. In the workflow editor, click **Step Templates** in the top right corner of the window.
 2. Click the external step and drag it into the workflow editor. Place the step where you want it.
 3. **Optional:** If you want to rename the step, right-click the step. Select **Properties**, and then click **General**. For the **Step name** property, type a name for the new step and click **OK**.
4. Edit the properties for processing behavior as needed.
5. Connect the step to other steps.

The external step can use conditional processing to receive a job from multiple steps and send it to multiple steps. You can attach rules with conditions to the connectors, and you can set job

properties for different branches of the workflow by adding steps based on the **AssignJobValues** step template.

6. Add or update the other steps in the workflow if necessary. A workflow can contain more than one step that calls an external program.
7. Save and enable the workflow.
8. Test the external program.

4. Using Web Services

- Using RICOH ProcessDirector REST API documentation
- Preparing to submit jobs using web services

If your application provides a different interface for displaying information about the RICOH ProcessDirector system, or if it needs to extract information from RICOH ProcessDirector to do more processing, you can use web services to request different kinds of information and manage system objects.

Preparing RICOH ProcessDirector to receive web service requests

Before you can submit web services requests, you must create a RICOH ProcessDirector user and password for web services to use. Be sure that the user ID has the appropriate level of authority in RICOH ProcessDirector to do the actions provided in the web service. For the examples in this chapter, **Operator** authority is sufficient. If you are using Lightweight Directory Access Protocol (LDAP) for authentication, create a user in LDAP for web services to use. The LDAP user must belong to a group that has the appropriate level of authority in RICOH ProcessDirector.

Because all users must change their passwords the first time they log in, log in to the RICOH ProcessDirector user interface using the user ID that you created and change the password. If you have passwords set to expire after a given time period, you must log in and change this password as needed.

Preparing web service requests

When you create requests, keep these points in mind :

- The web services user must log in before doing any actions.
To submit any web service requests, you must first use the **GET /users/login** web service to log in to RICOH ProcessDirector as the web service user and receive a credential token. That token must be included in the header of all the web service requests that you submit until you submit the **POST /users/logout/{name}** web service.
- Property names must be expressed in the database format, not in the user interface format.
When you include a property name in a request, you must use the database property name. For example, if you want to use **GET /objects/{objectType}** to retrieve the **Class** value of a job, you insert `Job.Class` as the **attribute** value in the request. Database property names are available in the field help for each property, in the product help system, in results received from some of the web services, and in [Database property names, p. 123](#).
- Property values must be expressed in the database format, not in the user interface format.
When you include a value in a query, you must make sure that it is a valid value. Usually, numerical fields require numerical values and text fields require text values. However, some properties have limited lists of values.

Any property that provides a drop-down list can only accept certain values. In addition, the values that appear in those lists are not always the same values that are stored in the database. For example, in the user interface, the values for the **Staple** property include:

- 2 at left
- 2 at right
- Top left
- Top left vertical
- Bottom left

However, the database values are:

- 2_at_left
- 2_at_right
- Top_left
- Top_left_vertical
- Bottom_left

In web service requests, you must use the database value. The database values for properties are available in the product help system and in [Database property names, p. 123](#).

- Positional job properties must include the phase, step, and workflow names in that order following the property name. Enclose each of the three names in brackets.

This example JSON string specifies a control file on a Linux system as the value of the **Identify PDF control file** job property (database name **Job.IdentifyPDFControlFile**):

```
"Job.IdentifyPDFControlFile[Prepare][IdentifyPDFDocuments][PullPDFSample]": "/aiw/aiw1/testfiles/PullPDF.ct1"
```

The job property is on the **IdentifyPDFDocuments** step in the **Prepare** phase of the **PullPDFSample** workflow.

4

Using RICOH ProcessDirector REST API documentation

RICOH ProcessDirector provides REST APIs for web service integration. Interactive documentation is provided with these APIs and provides live testing and information for each available API.

The REST API documentation includes many APIs that an application can use to integrate with RICOH ProcessDirector. You can use the REST API documentation and testing interface to test your parameter settings. When you use the interface to test web services, you access the RICOH ProcessDirector primary server. Actions such as creating, deleting, and enabling objects are executed on the primary server and are reflected in the RICOH ProcessDirector user interface. As a result, we suggest creating objects specifically for testing purposes.

These steps demonstrate how to request the log files for an object using REST APIs. These steps log you in to RICOH ProcessDirector, request the log files for the Sample printer, and log you out.

1. Open a Web browser and enter your RICOH ProcessDirector hostname or IP address into the address bar. Add `/restapi/` to the end of your hostname or IP address to access the REST API documentation. For example, `http://hostname:15080/restapi/`
2. In the **users** section, click **Expand Operations** and find **GET /users/login**.
3. Log in to RICOH ProcessDirector by entering your RICOH ProcessDirector credentials:
 - For the **name** parameter, enter your RICOH ProcessDirector user name.
 - For the **pwd** parameter, enter the password associated with your RICOH ProcessDirector user name.
4. Click the **Try it Out!**
The REST API documentation uses the values you enter to build a Curl command and a Request URL. Then it submits the request to the request URL and logs you in.

RICOH ProcessDirector sends a response indicating whether you successfully logged in. The response includes additional information including the token and the actions that this User ID is allowed to do.

- In the **Response Body** box, find and copy the token value.



```

{
  "token": "5273775233903331171",
  "actions": [
    "ViewGroupAttributes",
    "CopyGroup",
    "DeleteGroup",
    "ViewGroupLog",
    "MarkGroupForExport",
    "ExportGroup",
    "ExportGroupLogs",
    "UnmarkGroupForExport",
    "EditGroupAttributes",
    "ViewWorkflowBuilderTab",
    "WorkflowSystem.ImportFeature",
    "ViewArchiveTab",
    "ViewMainTab",
    "ConfigureMediaSettings",
    "Refresh",
    "ResetLayout",
  ]
}

```

- In the **objects** section, click **Expand Operations**. Scroll down until you find **GET /objects/log/{objectType}/{name}**.
- Retrieve the log messages for the Sample printer by entering these parameters:
 - For the **token** parameter, paste the token you copied above.
 - For the **objectType** parameter, type Printer.
 - The values for objectType are case-sensitive. You can use the web service **GET /util/objecttypes** to get a list of object types available on your system. The web service **GET /util/objecttypes** is found in the **util** section.
 - For the **name** parameter, type Sample.
- Click the **Try it Out!**
The log entries for the Sample printer are returned in the Response Body box.
The Curl and Request URL values are also returned.
- In the **users** section, click **Expand Operations** and find **POST /users/logout/{name}**.
- Log out of RICOH ProcessDirector:
 - For the **token** parameter, paste the token you copied above.
 - For the **name** parameter, enter your RICOH ProcessDirector user name.
- Click the **Try it Out!**
You are logged out of RICOH ProcessDirector.

Example values are included with certain APIs. With example values you can copy a sample string of code into the body field and modify it for testing.

These steps demonstrate how to use a REST API to connect a hot folder. These steps log you in to RICOH ProcessDirector, set the Input Device to accept files from **HotFolderPDF**, and log you out.

12. Open a Web browser and enter your RICOH ProcessDirector hostname or IP address into the address bar. Add `/restapi/` to the end of your hostname or IP address to access the REST API documentation. For example, `http://hostname:15080/restapi/`
13. In the **users** section, click **Expand Operations** and find **GET /users/login**.
14. Log in to RICOH ProcessDirector by entering your RICOH ProcessDirector credentials:
 - For the **name** parameter, enter your RICOH ProcessDirector user name.
 - For the **pwd** parameter, enter the password associated with your RICOH ProcessDirector user name.

15. Click the **Try it Out!**

The REST API documentation uses the values you enter to build a Curl command and a Request URL. Then it submits the request to the request URL and logs you in.

RICOH ProcessDirector sends a response indicating whether you successful logged in. The response includes additional information including the token and the actions that this User ID is allowed to do.

16. In the **Response Body** box, find and copy the token value.
17. In the **objects** section, click **Expand Operations**. Scroll down until you find **POST /objects/{objectType}/connect**.
18. Connect **HotFolderPDF** by entering these parameters:
 1. For the **token** parameter, paste the token you copied above.
 2. For the **objectType** parameter, type `InputDevice`.
The values for **objectType** are case-sensitive. You can use the web service **GET /util/objecttypes** to get a list of object types available on your system. The web service **GET /util/objecttypes** is found in the **util** section.
 3. For the body parameter, click the Example Value. The text inside the Example Value is copied into the body parameter. Replace the word “string” with “HotFolderPDF”. Make sure the double quotes are around the name of the object.
19. Click the **Try it Out!**
The properties and settings for HotFolderPDF are returned in the Response Body box. The Response Code and Response Headers are also returned.
20. In the **users** section, click **Expand Operations** and find **POST /users/logout/{name}**.
21. Log out of RICOH ProcessDirector:
 - For the **token** parameter, paste the token you copied above.
 - For the **name** parameter, enter your RICOH ProcessDirector user name.
22. Click the **Try it Out!**
You are logged out of RICOH ProcessDirector.

Preparing to submit jobs using web services

RICOH ProcessDirector provides a REST web service utility that you can use to submit jobs. If you can configure your application to use REST web services, you can use the **submitFile** utility to submit a file to a workflow or hot folder.

Depending on the requirements of your web service application, you need either a **Curl** command or **Request URL** to invoke the web service. Use the RICOH ProcessDirector REST API documentation to determine the correct syntax.

To prepare to submit a job using a web service:

1. Decide how RICOH ProcessDirector receives the file for processing. You can submit a file to RICOH ProcessDirector using a hot folder input device or you can submit directly to a workflow.
 - Submit files directly to a workflow when there is only one input file per job or you do not need to use the batching functions available in RICOH ProcessDirector.
 - Submit files to hot folders when you want to use the batching functions in RICOH ProcessDirector. You can only submit one file at a time with the web service. If a job has multiple input files, including job tickets, list files, or other resources, you must invoke the web service for each file.
Note the name of the workflow or hot folder.
2. Evaluate your web service application and see what information the application requires to submit a file to RICOH ProcessDirector. The RICOH ProcessDirector REST API documentation provides both a **Curl** command and a **Request URL**. Either can be used depending on what your web service application requires.
3. Use the RICOH ProcessDirector REST API documentation for the **submitFile** utility to help create the web service calls to submit your file. Open a web browser and enter your RICOH ProcessDirector hostname or IP address into the address bar. Add `/restapi/` to the end of your hostname or IP address to access the REST API documentation. For example: `http://hostname:15080/restapi/`.
Open the **util** section and find **POST /util/ submitFile/ {objectType}/ {name}**.
4. Update your web service application with the command or URL that you developed in previous step. Use the application to submit one or more files to RICOH ProcessDirector.
5. Verify that your web service application is configured correctly to submit a file to RICOH ProcessDirector.
 - If you submit to a hot folder, verify that the file is in the folder location in the hot folder. Connect and disable the hot folder if you want to check that the files are received in the correct directory without actually submitting the job. If you want the input device to submit the job, make sure it is both enabled and connected when your application submits the file.
 - If you submit to a workflow, verify that a job is created with the file you submitted. Make sure the workflow is enabled when you submit the file.

5. RICOH ProcessDirector symbol notation

You can use RICOH ProcessDirector symbol notation in formulas to describe the information source that RICOH ProcessDirector evaluates to set the value of a job property.

Symbol syntax

In RICOH ProcessDirector, this is the basic syntax of a symbol, which you can use in symbol formulas:
`${Name}`

Name is a database property name in RICOH ProcessDirector or a parameter that is passed with the job. RICOH ProcessDirector evaluates parameters that are passed with a job through a control file such as a rules file that parses JCL parameters and values. Name can also be a method in RICOH ProcessDirector, such as **getFileName**, **getAbsoluteFileName**, **getControlFileName**, or **getChildFileName**. The Name value is case-sensitive.

Examples

`${Get*Method}`

This symbol causes RICOH ProcessDirector to call an internal method and return an evaluated value. To see how RICOH ProcessDirector can use this type of symbol, see the **RunExternalProgram** step in the Prepare phase of the PDF workflow. The **RunExternalProgram** step specifies this **External Command** property:

```
cp ${getControlFileName()} /aiw/aiw1/samples/${Job.ID}.info.csv on AIX or Linux or  
copy ${getControlFileName()} C:\aiw\aiw1\samples\${Job.ID}.info.csv on Windows
```

The `${getControlFileName()}` symbol instructs RICOH ProcessDirector to copy the external control file template into the tmp subdirectory of the spool directory and resolve any symbols that the control file contains.

The **RunExternalProgram** step in the Prepare phase of the PDF workflow specifies this **External control file template** property:

```
/aiw/aiw1/control_files/external_programs/job_info.cfg on AIX or Linux or C:\aiw  
\aiw1\control_files\external_programs\job_info.cfg on Windows
```

When RICOH ProcessDirector creates the job that uses the workflow and sets its initial property values, it uses the value that the **getControlFileName** method returned to resolve the external command to this:

```
cp /aiw/aiw1/spool/default/10000003/tmp/job_info.control.text /aiw/aiw1/  
samples/10000003.info.csv on AIX or Linux or copy C:\aiw\aiw1\spool\default  
\10000003\tmp\job_info.control.text C:\aiw\aiw1\samples\10000003.info.csv on  
Windows
```

`${Job.PropertyName}`

This symbol causes RICOH ProcessDirector to query its database for the value of a specific RICOH ProcessDirector job property.

To see how RICOH ProcessDirector can use this type of symbol, review the contents of the **job_info.cfg** control file template. This control file template is in /aiw/aiw1/control_files/external_programs on AIX or Linux or C:\aiw\aiw1\control_files\external_programs on Windows.

When RICOH ProcessDirector creates a control file from this template, it resolves the values for the `${Job.ID}`, `${Job.Name}`, `${Job.JobType}`, `${Job.SubmitTime}`, `${Job.RequestedPrinter}`, `${Job.InputFile.Size}`, `${Job.JobSize}`, `${Job.TotalPages}`, and `${Job.TotalSheets}` symbols to the actual RICOH ProcessDirector job properties. For example:

```
`${Job.ID}`=10000001  
`${Job.Name}`=Demo.pdf
```

```

${Job.JobType}=PDF
${Job.SubmitTime}=10:02.35
${Job.RequestedPrinter}=Sample
${Job.InputFile.Size}=2608549
${Job.JobSize}=26
${Job.TotalPages}=26
${Job.TotalSheets}=26

```

↓ Note

- You can also use any of the system properties in a symbol formula that RICOH ProcessDirector evaluates. For example, **`${WorkflowSystem.Transform.Server.Address}`**.

`Math`

This symbol causes RICOH ProcessDirector to add, subtract, multiply, or calculate the modulus of two values which can be job properties or numbers. It can also generate a random number in a specified range.

The syntax of this symbol is:

```
Math(value1, operator, value2)
```

- value1* and *value2* are job properties in symbolic notation (such as `Job.CurrentTime`) or numbers. Numbers can contain fractional values, such as 2.45, if the property using the symbol formula supports floating point values.
- operator* is +, -, *, or mod for addition, subtraction, multiplication, and modulus respectively.

For example, to add 5 minutes to the current time and store it in the **Job.Info.Attr2** property, use `Math(Job.CurrentTime, +, 5)`.

Only properties that use integer, numeric, or timestamp values can be used in the *value* fields in this formula. Only properties that support symbol notation values can be set using this formula.

If either *value* is a timestamp property, the only operators supported are + and -, and the other value must be an integer. The units of the integer value are minutes.

Some properties appear to have timestamp values (such as `Job.TimeSubmitted`) but are defined as strings, so they cannot be used as values in a `Math` symbol. The `Math` symbol can be entered in fields on step templates in the Workflow Builder, including the **AssignJobValues** step template, and in fields in job property notebooks.

For modulus, *value1* must be an integer which is 0 or greater and *value2* must be an integer which is 1 or greater.

For random number generation, the syntax of this symbol is:

```
Math(rand, value1, value2)
```

For example, to generate a random number from 1 through 10, use `Math(rand, 1, 10)`.

↓ Note

- value1* and *value2* are job properties in symbolic notation (such as `Job.Copies`) or numbers.
- value1* and *value2* must be integers which are zero or greater.
- The result returned is an integer between *value1* and *value2*, inclusive.
- The random numbers that are generated are not cryptographically random.

`${RulesFileParameter}`

This symbol causes RICOH ProcessDirector to query a parameter file that accompanies an input file. It queries for a parameter value that is specific to another program or product.

For example, when the LPDPDF input device receives an input file, it uses the **receive_lpd_pdf_jobtype.cfg** control file to parse parameters that accompany the input file.

The **receive_lpd_pdf_jobtype.cfg** control file is located in this directory:

/aiw/aiw1/control_files/rules (Linux)

C:\aiw\aiw1\control_files\rules (Windows)

A **receive_lpd_pdf_jobtype.cfg** control file might contain this information:

```
orighost=mywindowshost
origuser=annsmith
origname=TestPDF.pdf
```

The symbols for these parameter values are:

```
${ORIGHOST}
${ORIGUSER}
${ORIGNAME}
```

To see how RICOH ProcessDirector can use this type of symbol, review the **receive_lpd_pdf_jobtype.cfg** file in this directory:

/aiw/aiw1/samples/rules (Linux)

C:\AIW\AIW1\samples\rules (Windows)

↓ Note

- See [Syntax for RICOH ProcessDirector control files, p. 105](#) for more detailed information about this implementation.

Usage notes for symbol formulas

These usage restrictions apply to using RICOH ProcessDirector symbol formulas:

Supported objects

RICOH ProcessDirector supports the use of symbol formulas only to set the values of job properties. You cannot use symbol formulas to set property values for any other object type, such as an input device or a printer. The symbol formula that RICOH ProcessDirector evaluates to set the value can be another job property, a primary server property, or a method call, such as: **\${Job.InputFile}**, **\${Printer.Model}**, and **\${getControlFileName()}**. However, a given symbol formula cannot contain both a job property and a system property or a method call at the same time.

↓ Note

- Changing the value of a system property might affect many symbol formulas for job properties. Because RICOH ProcessDirector updates all the symbol formulas at once, the operation can take a long time to complete.

Excluded properties

RICOH ProcessDirector does not support setting the value of the **Job.Class** property with a symbol formula.

Appearance in the RICOH ProcessDirector user interface

You can specify symbol formulas in workflows and in step templates through the Administration page of the RICOH ProcessDirector interface.

When viewing the property notebooks for these objects, symbol formulas always display in their formula format, such as `${Job.InputFile}` as the value for the **Job name** property. In the properties notebook for a job that uses a workflow with steps that specify symbol formulas, the affected job properties display their evaluated values from the formulas. For example, the value of the **Job name** property is the actual name of the input file, such as `reports.pdf`.

Multiple levels of formulas

The evaluation of formulas can extend to a group of related formulas. For example:

```
Job.Name=${Job.Description}
Job.Description=${Job.CustomerName}
Job.CustomerName=${Job.Locations}
```

In this case, when the **Job.Locations** job property has a value, RICOH ProcessDirector sets the value of the **Job.CustomerName** property. This, in turn, lets RICOH ProcessDirector set the value of the **Job.Description** property, and then set the value of the **Job.Name** property.

5

Circular formulas

A circular formula is one in which a property receives a value from a symbol formula, and then is used to provide a value for another related property. For example:

```
Job.Name=${Job.Description}
Job.Description=${Job.CustomerName}
Job.CustomerName=${Job.Name}
```

RICOH ProcessDirector does not support this usage and issues an error message.

Maximum depth for multiple levels of formulas

In the multiple levels of formulas example, the formula depth is three. RICOH ProcessDirector supports a depth of up to 99 related formulas. It issues an error message if it encounters a formula depth that is greater than 99.

Use of positional properties in symbol formulas to set values for non-positional job properties

Positional properties are properties that can appear in multiple phases and steps and that might have different values in each place. For example, an administrator could configure a workflow so that the **Valid return codes** job property on the **RunExternalProgram** step appears in multiple phases and steps in the workflow and has a different value each time. The phase and step names are the names of sections on the job property notebook with the individual property names and values for each instance shown in its own section. RICOH ProcessDirector does not let you use positional properties in symbol formulas that set values for non-positional properties because there is no mechanism to specify which occurrence of the positional property to use.

Use of positional properties in symbol formulas to set values for other positional job properties

Positional properties can use symbol formulas that specify other positional properties. RICOH ProcessDirector looks for the value of the positional property that it evaluates in the symbol formula in the same phase and step that the requesting positional property specifies.

Precedence of property values set by symbol formulas

When multiple methods that specify a value for the same property exist, RICOH ProcessDirector always uses the value that the symbol formula specifies. When a symbol formula exists for a property value, RICOH ProcessDirector does this:

- Discards any value specified in a control file.

For example, assume that the workflow specifies **`${Job.InputFile}`** as the value of the **Job name** property and a control file, such as `/aiw/aiw1/control_files/rules/` (AIX and Linux) or `C:\aiw\aiw1\control_files\rules\` (Windows) `receive_lpd_pdf_jobtype.cfg`, specifies: `DEFINE ${Job.Name} AS "${ORIGHOST}"`.

RICOH ProcessDirector sets the value of the **Job name** property to the name of the input file for the job, not to the value that the `ORIGHOST` parameter for the job specifies.

- Discards any explicitly specified value in the job properties notebook for a job property that the workflow defaults with a symbol formula. You must delete the symbol formula from the workflow and process the job again to use an explicitly specified value.

Validation of symbol formulas

RICOH ProcessDirector validates the syntax and the content of the symbol formula and issues messages for error conditions. For example, both of these would result in errors: **`Job.Description=${Job.CustomerName}`** and **`Job.Description=${Job.XYZ}`**. The first example is a syntax error with no closing brace and the second example is an unknown job property name.

6. File system mapping file for job tickets

The sample `system_map.cfg` found in `/aiw/aiw1/samples/config/` (AIX and Linux) or `C:\aiw\aiw1\samples\config\` (Windows) translates the file paths in JDF job tickets to the file paths of a mounted file system. You can copy this file into the `/aiw/aiw1/control_files/config/` (AIX and Linux) or `C:\aiw\aiw1\control_files\config\` (Windows) directory and edit it as necessary.

Each line of the file system mapping file is in this format:

```
client_file_path;host_file_path
```

client_file_path

The client file path is the file path as it appears in the job ticket. It must include at least one backslash (\) or slash (/) character. It can include an asterisk (*) as a wild card representing the drive letter.

host_file_path

The host file path is the file path where the RICOH ProcessDirector server can locate input files. It must include at least one backslash (\) or slash (/) character. It must not include wild cards.

On Linux, this line converts any Windows file path to a Linux file path:

```
*:\;/
```

Example

Suppose that the file system mapping file contains these lines:

```
C:\production\siteA;/BankFiles/prod  
*:\production\siteA\test;/BankFiles/test
```

The job ticket refers to a file called `C:\production\siteA\test\justAtest.pdf`. RICOH ProcessDirector searches for `justAtest.pdf` in these directories on AIX and Linux:

1. `/BankFiles/prod/test/`
2. `/BankFiles/test/`
3. `/production/siteA/test/`
4. The staging location of the hot folder input device

And in these directories on Windows:

1. `D:\BankFiles\prod\test\`
2. `D:\BankFiles\test`
3. The staging location of the hot folder input device

7. Syntax for RICOH ProcessDirector control files

- For rules
- For Passthrough printers
- For the Archive feature

Various parts of RICOH ProcessDirector use the information in control files or control file templates to set and pass values for different properties. You can copy the control files and control file templates that RICOH ProcessDirector provides and modify them to meet the needs of the installation.

Sample control files for rules

RICOH ProcessDirector provides sample control files for rules that parse JCL parameters, LPD control file parameters, or JDF values to set workflows and to set job property values.

The sample control files (`receive_jcl_jobtype.cfg`, `receive_lpd_jobtype.cfg`, `receive_lpd_pdf_jobtype.cfg`, and `receive_text_jobtype.cfg`) for rules are installed in the `/aiw/aiw1/samples/rules/` (AIX and Linux) or `C:\aiw\aiw1\samples\rules\` (Windows) directory.

To create your own control file, you can copy and rename one of the sample files to the `/aiw/aiw1/control_files/rules/` (AIX and Linux) or `C:\aiw\aiw1\control_files\rules\` (Windows) directory, then edit it to meet your needs.

Note

- Updates might overwrite files in the `/aiw/aiw1/samples/` (AIX and Linux) or `C:\aiw\aiw1\samples\` (Windows) directory, but they do not overwrite files in the `/aiw/aiw1/control_files/` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory. We recommend copying sample files into the `/aiw/aiw1/control_files` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory and making all your changes in the copied file.

7

receive_jcl_jobtype.cfg

The sample `receive_jcl_jobtype.cfg` file sets the workflow and job properties for jobs received from Download for z/OS and AFP Download Plus.

The AFP Support feature provides the `receive_jcl_jobtype.cfg` file.

RICOH ProcessDirector can use this control file to interpret a JCL file that accompanies a PRD dataset that RICOH ProcessDirector receives from a Download input device. For example, the JCL file might contain this information:

```
"-odatat=af -oburst=no -occ=yes -occtype=m -ocop=1 -odatac=unblock
-ofileformat=stream -of=F1HPSTP1 -ojobn=HPUNCH05 -ono=BLDPDEV9
-opr=HPUNCH -ous=WAITE
-opa=class=B,dest=LOCAL,forms=STD,jobid=JOB00105"
```

To use a control file, set the value of the **Child workflow initialization step** property for the input device to **SetJobTypeFromRules** or **SetJobTypeFromFileName**, and then set the value of the **Child workflow parsing rules** property to the path and file name of the control file. The **SetJobTypeFromRules** step uses the control file to set the workflow for the job, convert an optional overrides file submitted with a job to a file in RICOH ProcessDirector *property name=value* format for setting job properties, or both; the **SetJobTypeFromFileName** step uses the control file for setting job properties. All the information in the control file is case-sensitive.

 **Note**

You cannot use a control file to set job properties that are read-only in the Job Properties notebook.

The control file contains these sections:

CONFIGURATION section

This is a global settings section consisting of keywords that define how RICOH ProcessDirector interprets the JCL file.

FILE_MODE

This keyword controls how RICOH ProcessDirector processes the JCL file. A value of **"FILE"** instructs RICOH ProcessDirector to treat all the information in the file as a single record. In this mode, RICOH ProcessDirector can do search and replace actions. Always use the **"FILE"** value for control files that process JCL files. The double quotation marks in the value are required.

ATTRIBUTE_PATTERN

This keyword specifies a regular expression that defines how RICOH ProcessDirector recognizes the names of properties. As supplied by RICOH ProcessDirector, the value is **"\\${Job.*\}"**. The double quotation marks delimit the expression and the backslash characters are escape characters that precede the special characters in the expression.

The **"\\${Job.*\}"** value instructs RICOH ProcessDirector to recognize property names as strings that begin with **Job.** and that are followed by zero or more characters. RICOH ProcessDirector job property names match this convention, such as **Job.Duplex** and **Job.Print.CumulativeSheetsStacked**.

KEYWORD_CASE

This keyword defines the case of the characters in the JCL file parameter names. Depending on settings on the sending system, parameters might be passed as all uppercase or all lowercase characters. Use a value of **"UPPER"** or **"LOWER"** based on the requirements of the installation.

NORMALIZER_PROGRAM

This keyword specifies the name of a program that modifies the JCL file so all the properties it defines are in the form **"keyword=value"**. RICOH ProcessDirector supplies the **normalize_jcl.pl** program to modify the various styles of JCL parameters into the **"keyword=value"** form.

For example, consider this JCL string:

```
"-odatat=af -oburst=no -opa=class=B,dest=LOCAL,forms=STD,jobid=JOB00105"
```

When RICOH ProcessDirector uses the **normalize_jcl.pl** program specified in the control file, it replaces the JCL string with this value:

```
datat=af,burst=no,class=B,dest=LOCAL,forms=STD,jobid=JOB00105,
```

Delimit the beginning and ending of the **CONFIGURATION** section with **CONFIGURATION** and **ENDCONFIGURATION**.

REPLACE section

This section uses sed commands to replace strings in the JCL file. It is commented out in the sample file because the program defined by the **NORMALIZER_PROGRAM** keyword converts the JCL file.

For example, this statement replaces all occurrences of `-opa=` (note the initial space) with commas:

```
#s! -opa=!,!
```

Delimit the beginning and ending of the **REPLACE** section with **REPLACE** and **ENDREPLACE**.

PATTERN KEY_VALUE section

This section describes how RICOH ProcessDirector finds keywords and values, and converts them into tokens by using regular expression groups. As supplied by RICOH ProcessDirector, the section looks like this:

```
PATTERN KEY_VALUE
"(.*)=(.*)","
ENDPATTERN
```

The pattern is delimited by double quotation marks and the pattern to the left of the equals sign represents the keyword. The pattern to the right represents the value. This pattern creates a comma-delimited list of keyword and value pairs.

DEFINE statements section

This section uses symbol formulas to set the RICOH ProcessDirector workflow and job properties from the values for parameters that are passed in the JCL file with the job. Some examples of the types of **DEFINE** statements that the section can contain are:

```
DEFINE ${Job.JobType} AS "BILLS" WHEN (${DEST} == "LOCAL")
DEFINE ${Job.Class} AS "${CLASS}"
DEFINE ${Job.InputDatastream} AS "AFP" WHEN (${DATAT} == "af")
DEFINE ${Job.Destination} AS "${DEST}"
DEFINE ${Job.RequestedPrinter} AS "${DEST}"
DEFINE ${Job.Customer} AS "XYZ" WHEN (${CLASS} == "Z") FINALLY QUIT
```

The first **DEFINE** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.JobType** property to **BILLS** when the value of the **DEST** parameter from the JCL file is **LOCAL**. If the **DEST** parameter has any other value, RICOH ProcessDirector does not set the workflow from the control file. It sets it using another method, such as by using the workflow that is assigned to the Download input device.

The second **DEFINE** statement is a non-conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.Class** job property to the value of the **CLASS** parameter that is passed with the job. Therefore, if the original JCL string that RICOH ProcessDirector receives contains `-opa=class=B`, RICOH ProcessDirector sets the value of the **Job.Class** property to **B**.

The third **DEFINE** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.InputDataStream** job property to **AFP** when the value of the **DATAT** parameter from the JCL is **af**. If the **DATAT** parameter has any other value, RICOH ProcessDirector does not set the value of the **Job.InputDataStream** property.

The fourth and fifth **DEFINE** statements are non-conditional statements that set the values of two different job properties to the value of the same parameter.

The last **DEFINE** statement is a conditional statement that can cause RICOH ProcessDirector to stop reading any more **DEFINE** statements in the control file. If the condition that the statement defines is true, RICOH ProcessDirector stops reading the control file. If the condition is false, RICOH ProcessDirector continues to evaluate any other **DEFINE** statements that follow the **FINALLY QUIT** statement.

receive_lpd_jobtype.cfg

The sample `receive_lpd_jobtype.cfg` file sets the workflow and job properties for AFP jobs received through the LPD protocol.

The AFP Support feature provides the `receive_lpd_jobtype.cfg` file.

RICOH ProcessDirector can use this control file to interpret an LPD control file that accompanies an AFP print job received through the LPD print protocol. The format of the LPD control file depends on the operating system of the sending host. For example, an LPD control file received from Windows might contain this information:

```
orighost=mywindowshost
origuser=annsmith
origname=TestPDF.pdf
```

To use a control file, set the value of the **Child workflow initialization step** property for the input device to **SetJobTypeFromRules** or **SetJobTypeFromFileName**, and then set the value of the **Child workflow parsing rules** property to the path and file name of the control file. The **SetJobTypeFromRules** step uses the control file to set the workflow for the job, convert an optional overrides file submitted with a job to a file in RICOH ProcessDirector *property name=value* format for setting job properties, or both; the **SetJobTypeFromFileName** step uses the control file for setting job properties. All the information in the control file is case-sensitive.

Note

You cannot use a control file to set job properties that are read-only in the Job Properties notebook.

The control file contains these sections:

CONFIGURATION section

This is a global settings section consisting of keywords that define how RICOH ProcessDirector interprets the LPD control file parameters.

FILE_MODE

This keyword controls how RICOH ProcessDirector processes the LPD control file. A value of **"FILE"** instructs RICOH ProcessDirector to treat all the information in the file as a single record. In this mode, RICOH ProcessDirector can do search and replace actions. The double quotation marks in the value are required.

A value of **"RECORD"** for the file mode causes RICOH ProcessDirector to read the information in the file on a record-by-record basis. The double quotation marks in the value are required.

ATTRIBUTE_PATTERN

This keyword specifies a regular expression that defines how RICOH ProcessDirector recognizes the names of properties. As supplied by RICOH ProcessDirector, the value is **"\\${Job.*}"**. The double quotation marks delimit the expression and the backslash characters are escape characters that precede the special characters in the expression.

The **"\\${Job.*}"** value instructs RICOH ProcessDirector to recognize property names as strings that begin with **Job.** and that are followed by zero or more characters. RICOH ProcessDirector job property names match this convention, such as **Job.Duplex** and **Job.Print.CumulativeSheetsStacked**.

KEYWORD_CASE

This keyword defines the case of the characters in the LPD control file parameter names. Depending on settings on the sending system, parameters might be passed as all uppercase or all lowercase characters. Use a value of **"UPPER"** or **"LOWER"** based on the requirements of the installation.

Delimit the beginning and ending of the **CONFIGURATION** section with **CONFIGURATION** and **ENDCONFIGURATION**.

REPLACE section

This section uses sed commands to replace strings in the LPD control file. It is commented out in the sample file. You will probably not need to use it.

Delimit the beginning and ending of the **REPLACE** section with **REPLACE** and **ENDREPLACE**.

PATTERN KEY_VALUE section

This section describes how RICOH ProcessDirector finds keywords and values, and converts them into tokens by using regular expression groups. As supplied by RICOH ProcessDirector, the section looks like this:

```
PATTERN KEY_VALUE
"(. *?)=(. *?), "
ENDPATTERN
```

The pattern is delimited by double quotation marks and the pattern to the left of the equals sign represents the keyword. The pattern to the right represents the value. This pattern creates a comma-delimited list of keyword and value pairs.

DEFINE statements section

This section uses symbol formulas to set the RICOH ProcessDirector workflow and job properties from the values in the LPD control file that was passed with the job. Some examples of the types of **DEFINE** statements that the section can contain are:

Example for AIX:

```
DEFINE ${Job.JobType} AS "PDF" WHEN (${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Name} AS "${TITLETEXT}"
DEFINE ${Job.Host.UserID} AS "${ORIGUSER}"
DEFINE ${Job.InputDatastream} AS "PDF" WHEN
(${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Customer} AS "XYZ" WHEN
(${ORIGUSER} == "xyzadmin") FINALLY QUIT
```

Example for Linux:

```
DEFINE ${Job.JobType} AS "PDF" WHEN (${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Name} AS "${ORIGNAME}"
DEFINE ${Job.Host.UserID} AS "${ORIGUSER}"
DEFINE ${Job.InputDatastream} AS "PDF" WHEN
(${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Customer} AS "XYZ" WHEN
(${ORIGUSER} == "xyzadmin") FINALLY QUIT
```

The **DEFINE \${Job.JobType}** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.JobType** property to PDF when the value of the **ORIGHOST** parameter from the LPD control file is mywindowshost. If the **ORIGHOST** parameter

has any other value, RICOH ProcessDirector does not set the workflow from the control file. It sets it using another method, such as by using the workflow that is assigned to the LPD input device.

The **DEFINE \${Job.Name}** statement sets the value of the **Job.Name** property to the name of the original input file.

The **DEFINE \${Job.Host.UserID}** statement is a non-conditional statement. In this example, RICOH ProcessDirector sets the value of the **Job.Host.UserID** job property to the value of the **ORIGUSER** parameter in the LPD control file. Therefore, if the original LPD control file that RICOH ProcessDirector receives contains `origuser=annsmith`, RICOH ProcessDirector sets the value of the **Job.Host.UserID** property to `annsmith`.

The **DEFINE \${Job.InputDataStream}** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.InputDataStream** job property to PDF when the value of the **ORIGHOST** parameter from the LPD control file is `mywindowshost`. If the **ORIGHOST** parameter has any other value, RICOH ProcessDirector does not set the value of the **Job.InputDataStream** property.

The **DEFINE \${Job.Customer}** statement is a conditional statement that can cause RICOH ProcessDirector to stop reading any more **DEFINE** statements in the control file. If the condition that the statement defines is true, RICOH ProcessDirector stops reading the control file. If the condition is false, RICOH ProcessDirector continues to evaluate any other **DEFINE** statements that follow the **FINALLY QUIT** statement.

receive_lpd_pdf_jobtype.cfg

7

The sample `receive_lpd_pdf_jobtype.cfg` file sets the workflow and job properties for PDF jobs received through the LPD protocol.

RICOH ProcessDirector can use this control file to interpret an LPD control file that accompanies a PDF print job received through the LPD print protocol. The format of the LPD control file depends on the operating system of the sending host. For example, an LPD control file received from Windows might contain this information:

```
orighost=mywindowshost
origuser=annsmith
origname=TestPDF.pdf
```

To use a control file, set the value of the **Child workflow initialization step** property for the input device to **SetJobTypeFromRules** or **SetJobTypeFromFileName**, and then set the value of the **Child workflow parsing rules** property to the path and file name of the control file. The **SetJobTypeFromRules** step uses the control file to set the workflow for the job, convert an optional overrides file submitted with a job to a file in RICOH ProcessDirector *property name=value* format for setting job properties, or both; the **SetJobTypeFromFileName** step uses the control file for setting job properties. All the information in the control file is case-sensitive.

↓ Note

- You cannot use a control file to set job properties that are read-only in the Job Properties notebook.

The control file contains these sections:

CONFIGURATION section

This is a global settings section consisting of keywords that define how RICOH ProcessDirector interprets the LPD control file parameters.

FILE_MODE

This keyword controls how RICOH ProcessDirector processes the LPD control file. A value of **"FILE"** instructs RICOH ProcessDirector to treat all the information in the file as a single record. In this mode, RICOH ProcessDirector can do search and replace actions. The double quotation marks in the value are required.

A value of **"RECORD"** for the file mode causes RICOH ProcessDirector to read the information in the file on a record-by-record basis. The double quotation marks in the value are required.

ATTRIBUTE_PATTERN

This keyword specifies a regular expression that defines how RICOH ProcessDirector recognizes the names of properties. As supplied by RICOH ProcessDirector, the value is **"\\${Job.*}"**. The double quotation marks delimit the expression and the backslash characters are escape characters that precede the special characters in the expression.

The **"\\${Job.*}"** value instructs RICOH ProcessDirector to recognize property names as strings that begin with **Job.** and that are followed by zero or more characters. RICOH ProcessDirector job property names match this convention, such as **Job.Duplex** and **Job.Print.CumulativeSheetsStacked**.

KEYWORD_CASE

This keyword defines the case of the characters in the LPD control file parameter names. Depending on settings on the sending system, parameters might be passed as all uppercase or all lowercase characters. Use a value of **"UPPER"** or **"LOWER"** based on the requirements of the installation.

Delimit the beginning and ending of the **CONFIGURATION** section with **CONFIGURATION** and **ENDCONFIGURATION**.

REPLACE section

This section uses sed commands to replace strings in the LPD control file. It is commented out in the sample file. You will probably not need to use it.

Delimit the beginning and ending of the **REPLACE** section with **REPLACE** and **ENDREPLACE**.

PATTERN KEY_VALUE section

This section describes how RICOH ProcessDirector finds keywords and values, and converts them into tokens by using regular expression groups. As supplied by RICOH ProcessDirector, the section looks like this:

```
PATTERN KEY_VALUE
"(.*)=(.*)","
ENDPATTERN
```

The pattern is delimited by double quotation marks and the pattern to the left of the equals sign represents the keyword. The pattern to the right represents the value. This pattern creates a comma-delimited list of keyword and value pairs.

DEFINE statements section

This section uses symbol formulas to set the RICOH ProcessDirector workflow and job properties from the values in the LPD control file that was passed with the job. Some examples of the types of **DEFINE** statements that the section can contain are:

```
DEFINE ${Job.JobType} AS "PDF" WHEN (${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Name} AS "${ORIGNAME}"
DEFINE ${Job.Host.UserID} AS "${ORIGUSER}"
DEFINE ${Job.InputDatastream} AS "PDF" WHEN
(${ORIGHOST} == "mywindowshost")
DEFINE ${Job.Customer} AS "XYZ" WHEN
(${ORIGUSER} == "xyzadmin") FINALLY QUIT
```

The **DEFINE \${Job.JobType}** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.JobType** property to PDF when the value of the **ORIGHOST** parameter from the LPD control file is mywindowshost. If the **ORIGHOST** parameter has any other value, RICOH ProcessDirector does not set the workflow from the control file. It sets it using another method, such as by using the workflow that is assigned to the LPD input device.

The **DEFINE \${Job.Name}** statement sets the value of the **Job.Name** property to the name of the original input file.

The **DEFINE \${Job.Host.UserID}** statement is a non-conditional statement. In this example, RICOH ProcessDirector sets the value of the **Job.Host.UserID** job property to the value of the **ORIGUSER** parameter in the LPD control file. Therefore, if the original LPD control file that RICOH ProcessDirector receives contains origuser=annsmith, RICOH ProcessDirector sets the value of the **Job.Host.UserID** property to annsmith.

The **DEFINE \${Job.InputDatastream}** statement is a conditional statement. In the example, RICOH ProcessDirector sets the value of the **Job.InputDataStream** job property to PDF when the value of the **ORIGHOST** parameter from the LPD control file is mywindowshost. If the **ORIGHOST** parameter has any other value, RICOH ProcessDirector does not set the value of the **Job.InputDataStream** property.

The **DEFINE \${Job.Customer}** statement is a conditional statement that can cause RICOH ProcessDirector to stop reading any more **DEFINE** statements in the control file. If the condition that the statement defines is true, RICOH ProcessDirector stops reading the control file. If the condition is false, RICOH ProcessDirector continues to evaluate any other **DEFINE** statements that follow the **FINALLY QUIT** statement.

receive_text_jobtype.cfg

The sample receive_text_jobtype.cfg file sets the **Workflow** from a text overrides file.

The RICOH ProcessDirector-supplied **SetJobTypeFromRules** step can use the receive_text_jobtype.cfg control file to set the workflow from the jobID.overrides.text file. The step can also use this control file to convert an optional Job Definition Format (JDF) job ticket file, jobID.overrides.jdf, to a temporary text-based overrides file that it can use with the jobID.overrides.text file to set the workflow.

All the information in the control file is case-sensitive.

The control file contains these sections:

CONFIGURATION section

This is a global settings section consisting of keywords that define how RICOH ProcessDirector interprets the job ticket parameters.

FILE_MODE

This keyword controls how RICOH ProcessDirector processes the job ticket. A value of **"FILE"** instructs RICOH ProcessDirector to treat all the information in the file as a single record. In this mode, RICOH ProcessDirector can do search and replace actions. The double quotation marks in the value are required.

A value of **"RECORD"** for the file mode causes RICOH ProcessDirector to read the information in the file on a record-by-record basis. The double quotation marks in the value are required.

ATTRIBUTE_PATTERN

This keyword specifies a regular expression that defines how RICOH ProcessDirector recognizes the names of properties. As supplied by RICOH ProcessDirector, the value is **"\\${Job.*}"**. The double quotation marks delimit the expression and the backslash characters are escape characters that precede the special characters in the expression.

The **"\\${Job.*}"** value instructs RICOH ProcessDirector to recognize property names as strings that begin with **Job.** and that are followed by zero or more characters. RICOH ProcessDirector job property names match this convention, such as **Job.Duplex** and **Job.Print.CumulativeSheetsStacked**.

KEYWORD_CASE

This keyword defines the case of the characters in the job ticket parameter names. Depending on settings on the sending system, parameters might be passed as all uppercase or all lowercase characters. Use a value of **"UPPER"** or **"LOWER"** based on the requirements of the installation.

Delimit the beginning and ending of the **CONFIGURATION** section with **CONFIGURATION** and **ENDCONFIGURATION**.

REPLACE section

This section uses sed commands to replace strings in the job ticket. It is commented out in the sample file. You will probably not need to use it.

Delimit the beginning and ending of the **REPLACE** section with **REPLACE** and **ENDREPLACE**.

PATTERN KEY_VALUE section

This section describes how RICOH ProcessDirector finds keywords and values, and converts them into tokens by using regular expression groups. As supplied by RICOH ProcessDirector, the section looks like this:

```
PATTERN KEY_VALUE
"(.*)=(.*)","
ENDPATTERN
```

The pattern is delimited by double quotation marks and the pattern to the left of the equals sign represents the keyword. The pattern to the right represents the value. This pattern creates a comma-delimited list of keyword and value pairs.

DEFINE statements section

This section uses symbol formulas to set the RICOH ProcessDirector workflow from a value in the job ticket that was passed with the job. This is the type of **DEFINE** statement that RICOH ProcessDirector typically uses to set the workflow:

```
DEFINE ${Job.JobType} AS "Transform"
DEFINE ${Job.JobType} AS "PDF" WHEN (${Job.InputDatastream} == "pdf")
```

The first **DEFINE** statement sets the default workflow. The second **DEFINE** statement is conditional. In this example, RICOH ProcessDirector sets the value of the **Job.JobType** property to PDF when the value of the **Job.InputDatastream** parameter in the text overrides file is pdf. If the **Job.InputDatastream** parameter has any other value, RICOH ProcessDirector uses the default workflow.

Using the -outbin parameter in JCL and LPD jobtype files

The receive_jcl_jobtype.cfg, receive_lpd_jobtype.cfg, and receive_lpd_pdf_jobtype.cfg control files provided with RICOH ProcessDirector map the value of the -outbin parameter to the Job.OutputBin property.

To use the -outbin parameter in JCL and LPD jobtype files:


1. Fill in this table with information from your application, printers, and bins. Each row reflects information about one bin.

Column 1: Values used in -outbin parameter (property value)	Column 2: Name of the bin in Properties notebook (object name, outputBin name)	Column 3: Printer model (printerModel name)	Column 4: Printer bin number (binNumber)
Example: 9	Example: Stacker9	Example: InfoPrint 2085	Example: 9

2. Copy this text into a blank text file:

```
<IPPD_UpdateData version="1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <object name="Stacker9" type="OutputBin">
    <property name="OutputBin.BinNumber" value="9"/>
  </object>
  <printerModel name="InfoPrint 2085">
    <outputBin name="Stacker9" binNumber="9"/>
  </printerModel>
</IPPD_UpdateData>
```

3. Edit the text file with the values that you entered in the table:
 1. Copy the <object> and <printerModel> tag sets so you have one set for each row in the table.
 2. Use the values from column 1 for the value attribute of the property tag.

3. Use the values from column 2 for the name attribute of both the object and outputBin tags.
4. Use the values from column 3 for the name attribute of the printerModel tag.
5. Use the values from column 4 for the binNumber attribute of the outputBin tag.
4. Save the file.
5. Click the **Administration** tab.
6. In the left pane, click **Utilities** → **Import Objects**.
7. Click  and navigate to the XML file that you just created. Click **Open**.
8. Click **Import**.

Sample control file templates for Passthrough printers

RICOH ProcessDirector provides a sample control file template for use with Passthrough printers. This template is called `passthru.cfg` and is installed in: `/aiw/aiw1/samples/passthru/` (AIX and Linux) or `C:\aiw\aiw1\samples\passthru\` (Windows).

If the printer command that you specify for a Passthrough printer uses a control file, the control file that RICOH ProcessDirector generates from the control file template specifies options for the printer command. Authorized users can copy and modify the sample control file template that RICOH ProcessDirector provides. They then put the customized control file template in any directory that is accessible to RICOH ProcessDirector. Use the **Control file template** printer property to specify the name and location of the control file template.

Note

- Updates might overwrite files in the `/aiw/aiw1/samples/` (AIX and Linux) or `C:\aiw\aiw1\samples\` (Windows) directory, but they do not overwrite files in the `/aiw/aiw1/control_files` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory. We recommend copying sample files into the `/aiw/aiw1/control_files` (AIX and Linux) or `C:\aiw\aiw1\control_files` (Windows) directory and making all your changes in the copied file.

The `passthru.cfg` control file template uses RICOH ProcessDirector symbol formulas to set printer command parameter values. These are examples of the entries in the control file template:

```
JobID=${Job.ID}
JobCopies=${Job.Copies}
PrinterId=${Printer.ID}
CustomerName=${Printer.CustomerName}
```

The keywords to the left of the equal sign are sample parameters.

Note

Not all printer commands can use control files. If yours can, it might use different parameter names.

The values to the right are symbol formulas for RICOH ProcessDirector job properties. Using **`JobCopies=${Job.Copies}`** as an example, RICOH ProcessDirector sets the value of the **JobCopies** parameter to the value of the **Job.Copies** job property. For example, if you wanted the original input filename preserved when the job was sent to the printer, you could substitute **`${Job.Inputfile}`** for **`${Job.ID}`** in the control file template entry for the **JobID** parameter.

Control files for the Archive feature

The RICOH ProcessDirector Archive feature can use an Associated properties file and a Document properties file as input to the **StoreInRepository** step. The Associated properties file defines properties you want to store in the repository that are associated with a job but are properties of other objects. The Document properties file is a tab-delimited file containing property values to store in the repository. Neither file is required for Archive processing, but they may be useful to capture data your company desires about the production history of a job in its workflow.

The Archive feature can also produce a file containing the properties stored about a job or document when it was written to the repository. That file is produced by the **ExportFromRepository** step template and is called the Export results file

Creating an associated properties file

You can create an associated properties file to specify one or more properties that are associated with a job, but that are properties of other objects. When a step based on the **StoreInRepository** step template runs, these properties and their values are stored in a repository along with job and document data. For example, you can store the model of the printer requested for a job or the color of the media specified to print a job.

Note

- You can also specify positional job properties in an associated properties file and store their values in a repository. You cannot select a positional job property as a value for the **Job properties to store** property on the **StoreInRepository** step. In a workflow, the values of positional properties can be different for different steps based on the same step template.

Positional job properties

To store values for a positional job property, you specify the property, the phase that the step is in, the internal name of the step with the property, and a property label. The syntax is:

```
Job_property[Phase][Step_identifier]:Property_label
```

For example, you have the Automated Verification feature, and you want to store the name of the barcode reader that the **ReadBarcodeData** step in the Insert phase uses to track the documents in a job through an inserter.

When you create the associated properties file, you type this line into a text editor:

```
Job.TrackAndTrace.BarcodeReader[Insert][ReadBarcodeData]:Property_label
```

The property label might be `Job.BarcodeReader`.

When the **StoreInRepository** step runs, RICOH ProcessDirector:

- Gets the value of the **Barcode reader** job property (database name **Job.TrackAndTrace.BarcodeReader**) for the **ReadBarcodeData** step in the **Insert** phase. This value might be **BarcodeReader1**.
- Stores the value of the **Barcode reader** property along with other information for the job and its documents in the repository.

To see whether a property on a step template is positional, click the ? icon and check the **Usage notes** in the help.

Properties associated with a job

To store the value for a property of another object, you must be able to create a chain of relationships to that property. The chain must start with a job property that specifies an object as a value. The next property in the chain must be a property of the object specified by the job property. The chain must end with the property whose values you want to store.

Although you can start the chain with any job property that specifies an object as a value, these job properties satisfy most needs.

Object	User interface name of job property	Database name of job property	Base product or feature
Barcode reader	Barcode reader	Job.TrackAndTrace.BarcodeReader	Automated Verification
Input device	None	Job.SourceInputDeviceName	Base product
Inserter	Inserter controller	Job.InsertSystem.ID	Inserter
Media	Media	Job.Media	Base product
Printer	Requested printer	Job.RequestedPrinter	Base product

The following line shows a simple version of the syntax for specifying associated properties:

```
Property_to_store@Job_property:Property_label
```

Note

- If you chose **Any printer** on the **PrintJobs** step, you cannot use the Job.RequestedPrinter database name. Replace that name with Job.PreviousPrinter.

The system starts at the colon and reads the properties from right to left. The number of properties in the chain can vary. An @ symbol separates the properties. To the right of the colon is a property label. The property label is required.

These examples give the user interface names of properties with the database names in parentheses. Use the database names when you create the associated properties file.

- You want to store values for the **Printer model** printer property (database name **Printer.Model.Specific**). You can chain the **Requested printer** job property (database name **Job.RequestedPrinter**) directly to the printer property.

When you create the associated properties file, you type this line into a text editor:

```
Printer.Model.Specific@Job.RequestedPrinter:Property_label
```

The property label might be Job.PrinterModel.

When the **StoreInRepository** step runs, RICOH ProcessDirector:

- Gets the value of the **Requested printer** property.
This value might be **Printer4**.

2. Uses the **Printer.** portion of the **Printer.Model.Specific** property to identify the next object in the chain: a printer object.
3. Gets the value of the **Printer model** property for **Printer4**.
This value might be **Ricoh Pro C901**.
4. Stores the value of the **Printer model** property along with other information for the job and its documents in the repository.

★ Important

- The link between the job property and the object portion of the next property is critical. You must link the **Requested printer** job property to a printer property. The database name of a printer property starts with **Printer**. An example at the end of this topic shows how to link a job property to another property through an intermediate property.
- You have the Automated Verification feature, and you want to store values for the **Barcode Format** property (database name **BarcodeReader.BarcodeFormat**). You can chain the **Barcode reader** job property (database name **Job.TrackAndTrace.BarcodeReader**) directly to the **Barcode Format** property. Because the **Barcode reader** property is positional, you need to specify the Phase and Step identifier.

You have two different steps that read barcodes, and the barcode readers in the two steps use a different barcode format. You want the barcode reader that the **ReadBarcodeData** step in the Insert phase uses.

When you create the associated properties file, you type this line into a text editor:

```
BarcodeReader.BarcodeFormat@Job.TrackAndTrace.BarcodeReader[Insert]
[ReadBarcodeData]:Property_Label
```

The property label might be `Job.BarcodeFormat`.

When the **StoreInRepository** step runs, RICOH ProcessDirector:

1. Gets the value of the **Barcode reader** property for the **ReadBarcodeData** step in the **Insert** phase.
This value might be **BarcodeReader2**.
2. Uses the **BarcodeReader.** portion of the **BarcodeReader.BarcodeFormat** property to identify the next object in the chain: a barcode reader object.
3. Gets the value of the **Barcode Format** property for **BarcodeReader2**.
This value might be **BarcodeFormat2**.
4. Stores the value of the **Barcode Format** property along with other information for the job and its documents in the repository.

To create an associated properties file:

1. With a text editor, create a new file.
2. Type a line for the first property whose values you want to store.

Use this syntax:

```
Property_to_store@Intermediate_property@Job_property[Phase][Step_
identifier]:Property_Label
```

where:

- *Property_to_store* is the database name of the property you want to store.
- *Intermediate_property* is the database name of an intermediate property, if needed, that links the job property to the property you want to store by identifying an intermediate object, such as Media. If you need to specify two intermediate properties, separate them with an @ symbol.
You can link many job properties directly to properties you want to store without an *Intermediate_property*.
- *Job_property[Phase][Step_identifier]* has these parts:
 - *Job_property* is the database name of the job property that identifies an object such as a printer.
If you are storing a positional job property, which you cannot select as a value for the **Job properties to store** property on the **StoreInRepository** step, *Job_property* is the database name of the positional job property. You do not need to specify any additional properties.
 - If the property is positional, *Phase* is the name of the phase that the step is in, and *Step_identifier* is the internal name of the step with the property.
If the property is not positional, do not type a [*Phase*] or [*Step_identifier*].
- *Property_label* is the name that appears on the Properties tab when you click **Show details** on the Results table of the Archive tab. We recommend the format *Job.MyProperty*. The property label for each property in the associated properties file must be unique.

For example, you might type:

```
Job.TrackAndTrace.BarcodeReader[Insert][ReadBarcodeData]:Job.BarcodeReader
```

3. If you want to store values for a second property, type a line break, and then repeat the previous step for the second property.

For example, you might type:

```
Printer.Model.Specific@Job.RequestedPrinter:Job.PrinterModel
```

4. Save the text file.
For example, you might name the file `associatedproperties.txt`.
5. Send the associated properties file to the RICOH ProcessDirector server in a directory that the RICOH ProcessDirector system user has access to.

The file is now available to use as the value of the **Associated properties file** property for a **StoreInRepository** step in a workflow.

When the **StoreInRepository** step runs, RICOH ProcessDirector stores (with each document and job) the value of each stored property specified in the associated properties file.

- If the value is null for any stored property on a line of the associated properties file, RICOH ProcessDirector stores a null value for the property.
- If a property allows multiple selections, RICOH ProcessDirector stores the multiple selections separated by a vertical bar (|). For example, the value of a stored property might be **BarcodeReader1 | BarcodeReader2**.
- If multiple values are selected for both a job property and the associated object property, RICOH ProcessDirector adds an underscore and the name of the job property value to the property label. RICOH ProcessDirector stores each job property value separately. For example:

```
Job.BarcodeFormat_BarcodeReader1   Job.BarcodeFormat_BarcodeReader2
BarcodeFormat1|BarcodeFormat2       BarcodeFormat3
```

Users cannot search a repository for these properties. After you search on the Archive tab for job or document properties, RICOH ProcessDirector displays the values of associated properties on the Properties tab when you click **Show details** on the Results table.

Examples

Storing the model of the printer requested to print a job

- File contents:

```
Printer.Model.Specific@Job.RequestedPrinter:Job.PrinterModel
```

- Value of **Printer model** property:

```
Ricoh Pro C901
```

- Information stored in the repository for a specific job and each document in the job:

```
Job.PrinterModelRicoh Pro C901
```

- Information displayed in the properties notebook for the results of a search:
Job.PrinterModel: Ricoh Pro C901

↓ Note

- When processing an associated properties file, RICOH ProcessDirector does not store or display the value of the *Job_property_identifying_object* property or any *Property_identifying_intermediate_object* property.
- If you chose **Any printer** on the **PrintJobs** step, you cannot use the Job.RequestedPrinter database name. Replace that name with Job.PreviousPrinter.

Storing the barcode format used by a barcode reader

- File contents:

```
BarcodeReader.BarcodeFormat@Job.TrackAndTrace.BarcodeReader[Insert][ReadBarcodeData]:Job.B
```

- Barcode readers selected on the **ReadBarcodeData** step:

- **BarcodeReader1**
- **BarcodeReader2**

- Barcode formats selected for **BarcodeReader1**:

- **BarcodeFormat1**
- **BarcodeFormat2**

- Barcode format selected for **BarcodeReader2**: **BarcodeFormat3**.

- Information stored in the repository for a specific job and each document in the job:

```
Job.BarcodeFormat_BarcodeReader1   Job.BarcodeFormat_BarcodeReader2
BarcodeFormat1|BarcodeFormat2       BarcodeFormat3
```

- Information displayed in the properties notebook for the results of a search:
Job.BarcodeFormat_BarcodeReader1: BarcodeFormat1 | BarcodeFormat2
Job.BarcodeFormat_BarcodeReader2: BarcodeFormat3

Storing the reprint method used by an inserter

- File contents:

```
InserterSystem.ReprintMethod@Job.InserterSystem.ID:Job.InserterReprintMethod
```

- Value of **Reprint method** property (database name **InserterSystem.ReprintMethod**):

```
Open loop
```

- Information stored in the repository for a specific job and each document in the job:

```
Job.InserterReprintMethod  
Open loop
```

- Information displayed in the properties notebook for the results of a search:
Job.InserterReprintMethod: Open loop

Storing the folder location of the input device that received a job

- File contents:

```
InputDevice.FolderLocation@Job.SourceInputDeviceName:Job.InputDeviceFolder
```

- Value of **Folder location** property (database name **InputDevice.FolderLocation**):

```
/aiw/aiw1/System/hf/defaultPDF
```

- Information stored in the repository for a specific job and each document in the job:

```
Job.InputDeviceFolder  
/aiw/aiw1/System/hf/defaultPDF
```

- Information displayed in the properties notebook for the results of a search:
Job.InputDeviceFolder: /aiw/aiw1/System/hf/defaultPDF

Storing the color of the media requested to print a job

- File contents:

```
MediaType.Color@Media.MediaTypeID@Job.Media:Job.MediaColor
```

- Value of **Media color** property (database name **MediaType.Color**):

```
Ricoh Pro C901
```

- Information stored in the repository for a specific job and each document in the job:

```
Job.MediaColor  
Blue
```

- Information displayed in the properties notebook for the results of a search:
Job.MediaColor: Blue

Note

- You can replace the **Media color** property with other media type properties such as **Media weight** (database name **MediaType.Weight**) and **Media details** (database name **MediaType.Details**).

To store the values of a media size property, such as **Media height** (database name **MediaSize.Height**), use this line: `MediaSize.Height@Media.MediaSizeID@Job.Media:Job.MediaHeight`

Document properties file

The Document properties file contains values for document properties as well as properties that are not defined to RICOH ProcessDirector to be stored in a repository. The file can contain either or both types of properties that you want to store in the repository as part of your workflow. If the property is not defined as a RICOH ProcessDirector property, you cannot use it to retrieve a job or document from the archive, but you can see the value the job or document had for the property when it was stored.

If all of the property values to store are selected from the **Document properties to store** list, then the regular document properties file is used. If additional fields not known to RICOH ProcessDirector are used, then you must create a unique document properties file. This unique document properties file must be specified in the **Document properties file** property of the **StoreInRepository** step. Additionally, you must select ALL in the **Document properties to store** list to ensure that the values for the additional fields are saved in the document properties file.

The file requires a header row followed by value rows in a tab-delimited format like the regular document properties file. The data in the file might look like this: (spacing between entries has been altered for clarity)

Doc.Email.Address	Doc.Custom1.AwardLevel	Member since	Anvrsry date
John_Doe@mail.com	Gold	3_1999	03_15
Jane_Doe@mail.com	Silver	7_2009	07_23
Bob_Smith@mail.com	Tin	9_2013	09_14

7

Export results file

The export results file contains values for document properties that you have chosen to export from the repository as a comma-separated value (CSV) file. The file is created by the **ExportFromRepository** step.

The properties to be exported are selected by a query as specified on the **ExportFromRepository** property notebook. The query can either be drawn from a file or from direct entry in the **Search criteria** property. You must choose either **File** or **Text** in the **Criteria type** property to specify the source of the query. The query must appear in the same format as the search criteria on the **Results** portlet on the Archive tab.

If more than one **ExportFromRepository** step is used in a workflow, the additional results overwrite the results from any previous **ExportFromRepository** step unless you specify different results files in the **Export results file** property of subsequent **ExportFromRepository** steps.

The export results file contains a header row followed by value rows in a comma-delimited format. The headers and the values are all enclosed in double quotes. Null value returns are indicated by a dash surrounded by double quotes ("-"). The data in the file might look like this:

```
"Doc.OriginalSequence", "Doc.OriginalFirstPage", "Job.TotalPages"
"1", "1", "4186"
"2", "5", "4186"
"4", "13", "4186"
```

8. Database property names

- For jobs
- For workflows
- For printers
- For media
- For input devices
- For input files
- For Notification
- For servers
- For step templates
- For the system
- For users
- For groups
- For locations
- For security

RICOH ProcessDirector messages might refer to properties by their database names instead of the field names that appear in property notebooks. Most of the database property names are similar to the names in the property notebooks, but they are written in a different format.

When you install RICOH ProcessDirector features, you add additional database properties. Lists of these database property names are found in the Information Center for RICOH ProcessDirector.

Database property names for jobs

Some messages about jobs refer to job properties by their database names, which begin with **Job**. You can use the database property names for job properties in symbol formulas that you specify for RICOH ProcessDirector external programs. You can also specify symbol formulas for job properties in RICOH ProcessDirector control files.

Some of the values that you see in lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that you can change the value after the job has been submitted.
- No means that you cannot change the value.

In the **Job ticket** column:

- Yes means that the property can be set from one or more values in the job ticket that is used to submit the job.
- No means that the property cannot be set from values in the job ticket.

Job properties

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Add. BlankPage	Add blank page	Adds a blank page at the end of each PDF file with an odd number of pages when combining PDF files contained in a ZIP file.	<ul style="list-style-type: none"> • No • Yes 	Yes	No
Job.Collate	Print: Collation	Specifies the type of collation that the printer uses when printing multiple copies of the job.	<ul style="list-style-type: none"> • Not set • Off • Collate 	Yes	No
Job.Binding	Binding	Set the binding settings for the job you are printing.	<ul style="list-style-type: none"> • None • Perfect • Ring • Ring and punch 	Yes	Yes
Job.Cjfx. FailOnMissingX- path	XML: Stop when no matching elements	Specifies whether a step based on the CreateJobs-FromXML step template puts a job in the error state when no elements match the XPath expression.	<ul style="list-style-type: none"> • No • Yes 	Yes	No
Job.Cjfx. FileToSplit	XML: XML input file	Specifies the name of the XML file that a step based on the CreateJobs-FromXML step template uses as input to create jobs.		Yes	No
Job.Cjfx. TypeOfJob	XML: Create as child job	Specifies whether a step based on the CreateJobs-FromXML step template creates child jobs or independent jobs from the original job.	<ul style="list-style-type: none"> • No • Yes 	Yes	No
Job.Cjfx. Workflow	XML: Workflow for new jobs	Specifies the workflow that a step based on the		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		CreateJobs-FromXML step template submits new XML jobs to.			
Job.Cjfx.XPath	XML: XPath expression to create jobs	Specifies an XPath expression that identifies an element. Each time a step based on the CreateJobs-FromXML step template finds a matching element in the XML input file, it creates an XML file and submits it as a job.		Yes	No
Job.CompressAllFiles	Job Defaults tab of RetainCompleted-Jobs step template property notebook: Compress all files	Specifies whether to compress all spool and checkpoint files for the job when the job is retained.		Yes	No
Job.CompressFilePatterns	Job Defaults tab of CompressFiles step template property notebook: Compress file patterns	Specifies the job files that the step compresses.		Yes	No
Job.Copies	General and Status: Job copies requested	Contains the number of job copies that have been requested.		Yes	Yes
Job.CopiesStacked	Status: Job copies stacked	Contains the current number of job copies that have completed printing and that have reached the output stacker of the printer device.		No	No
Job.CreateJobFromFiles.JobType	Create Job: Workflow	Specifies the workflow to use for the child job.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job. CreateJobFrom- Files.JobName	Create Job: Job name	Specifies the name of the child job.		Yes	No
Job. CreateJobFrom- Files.DestUsage	Create Job: Group ID	Specifies the role of the destination file.		Yes	No
Job. CreateJobFrom- Files.DestType	Create Job: Group size	Specifies the destination file type.		Yes	No
Job. CreateJobFrom- Files.Source1	Create Job: First source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source2	Create Job: Second source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source3	Create Job: Third source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source4	Create Job: Fourth source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source5	Create Job: Fifth source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source6	Create Job: Sixth source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source7	Create Job: Seventh source file	Specifies the full path and name of the input file used to create the job.		Yes	No
Job. CreateJobFrom- Files.Source8	Create Job: Eighth source file	Specifies the full path and name of the input file used to create the job.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job. CurrentPrinter	Status: Assigned printer	Contains the name of the printer that RICOH ProcessDirector has assigned to print this job.		No	No
Job. CustomerName	Scheduling: Customer name	Identifies the customer who is associated with this job.		Yes	Yes
Job.Description	General: Job description	Contains text that describes the job.		Yes	Yes
Job. DownloadFile	URL for download file	Specifies the URL of the file to download.		Yes	No
Job. DownloadFile- ToCreate	Path to downloaded file	Specifies the file name and location where RICOH ProcessDirector saves the file.		Yes	No
Job.Duplex	General: Duplex	Indicates whether duplexed printing is active for the job and, if so, the type of duplexed printing.	<ul style="list-style-type: none"> • No • Tumble • Yes 	Yes	Yes
Job.EJB. SMTPserver	SMTP server type	Specifies whether you want this step to use the system default email server or an alternate email server.	<ul style="list-style-type: none"> • System • Alternate 	Yes	No
Job. EmailAd- dressBCC	Blind copy address	One or more email addresses to send a blind copy of the email to.		Yes	No
Job. EmailAddressCC	Copy address	One or more email addresses to send a copy of the email to.		Yes	No
Job. EmailAddressTo	Recipient address	One or more email addresses to send the job to.		Yes	No
Job. EmailMessage	Message	Specifies text to include in the body of the email.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Email. PageRange	Page range to send	Shows a numeric string that describes which pages in the print file are extracted, made into a separate file, and attached to the email.		Yes	No
Job.Email. RangeData Stream	Page range data stream	Specifies the data stream of the print file used to create the file with the page range indicated in the Page range to send property.	<ul style="list-style-type: none"> • Use current • AFP • PDF 	Yes	No
Job.Email.Secure Connection	Secure connection	Specifies whether the connection with the mail server should use SSL or TLS security.	<ul style="list-style-type: none"> • None • SSL • TLS 	Yes	No
Job. EmailSendFrom	Sender address	The email address used in the From: field of the email.		Yes	No
Job.EmailSubject	Subject line	Specifies the text to include in the subject line of the email.		Yes	No
Job.External. CodePage	External: External program code page	Specifies the code page to use to write the contents of the files that are sent to the external command for processing, such as the control file template. This code page is also used to read messages that the external command writes to stderr and stdout, as well as other files that the external command might create.	<ul style="list-style-type: none"> • UTF-8 • ISO8859_1 • ISO8859_15 • EUC_JPEUC_JP 	Yes	No
Job.External. Command	External: External command	Specifies a command string that		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		a step can execute during processing.			
Job.External. ControlFileTem- plate	External: External control file template	Specifies the path and name of the control file template that the external command uses.		Yes	No
Job.External. Language	External: External program language	Specifies what language the external command should use when it returns messages to RICOH ProcessDirector.	<ul style="list-style-type: none"> • de_DE • en_US • es_ES • fr_FR • it_IT • ja_JP 	Yes	No
Job.External. ValidRCs	External: Valid return codes	Lists return code values that the external command can issue, that indicate that the command ran successfully. You can specify multiple return code values by separating the values with commas. For example, 0, 4.		Yes	No
Job.FileToEmail	Attachments	Specifies the full paths to one or more files that should be attached to the email and sent to the recipients.		Yes	No
Job.FoldOptions	Fold Options	Specifies how to fold the job or large sheets in the job, whether to fold all the sheets together (overlap or collate), and whether the front side of the sheet ends up on the outside, rather than the inside, of the fold.	<ul style="list-style-type: none"> • None • Z fold • Z fold for large paper • Double parallel fold • Gate fold • Letter fold in • Letter fold out • Saddle • Saddle down 	Yes	Yes
Job. HeaderConfig	Header page configuration file	Specifies the path to and the name of the		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		configuration file that RICOH ProcessDirector uses to create the content and format of the header page placed before each file that is contained in the ZIP file. Authorized users can specify one of the configuration files for header pages that RICOH ProcessDirector provides, or specify a customized configuration file.			
Job.HotFolder. ApplicationLog- File	Hot Folder: Application log file	The directory where the external program stores its log files. You can specify this property.		Yes	No
Job.HotFolder. FileToSend	Hot Folder: File to send	Names the file that RICOH ProcessDirector puts in the sending folder to submit to the external program. You can edit this property.		Yes	No
Job.HotFolder. FileVerification- Count	Hot Folder: File verification count	The number of times that RICOH ProcessDirector polls the retrieval folder and finds that the size of the retrieved file has not changed before deciding that the retrieved file is complete.		Yes	No
Job.HotFolder. PollInterval	Hot Folder: Poll interval	The time between 2 consecutive polls of the retrieval folder. The unit of time for the value can be		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		seconds, minutes, or hours.			
Job.HotFolder. RetrievalFolder	Hot Folder: Retrieval folder	Names the output hot folder for the external program. RICOH ProcessDirector looks in the retrieval folder for the retrieved file using the retrieval pattern. You can edit this property.		Yes	No
Job.HotFolder. RetrievalPattern	Hot Folder: Retrieval pattern	The pattern-matching string that RICOH ProcessDirector uses to identify a returned job in the retrieval folder.		Yes	No
Job.HotFolder. RetrievedFile	Hot Folder: Retrieved file	The name to use to rename the retrieved file.		Yes	No
Job.HotFolder. SendingFolder	Hot Folder: Sending folder	Names the hot folder where RICOH ProcessDirector puts the job file to send to the external program.		Yes	No
Job.HotFolder. TimeOutInterval	Hot Folder: Timeout interval	The time in minutes before a job goes into an error state when the retrieved file is not found or not complete.		Yes	No
Job.ID	Displays in the properties notebook title.	Contains a unique number that identifies the job on the system.		No	No
Job.Info.Attr1	Information: Custom 1	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Info.Attr2	Information: Custom 2	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr3	Information: Custom 3	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr4	Information: Custom 4	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr5	Information: Custom 5	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr6	Information: Custom 6	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr7	Information: Custom 7	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr8	Information: Custom 8	Contains information about the job that is not included in other		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		properties because it is specific to your company or your processes.			
Job.Info.Attr9	Information: Custom 9	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job.Info.Attr10	Information: Custom 10	Contains information about the job that is not included in other properties because it is specific to your company or your processes.		Yes	No
Job. InputDatastream	General: Input data stream	Specifies the format of data that the input file for this job contains.	<ul style="list-style-type: none"> • afp • gif • jdf • jpeg • json • lcds • linedata • metacode • pcl • pdf • ps • text • tiff • unknown • xml 	No	Yes
Job.InputFile	General: Input file name	Contains the name of the original input file from which RICOH ProcessDirector created the job.		No	Yes
Job.InputFile.Size	General: Input file size (bytes)	Specifies the size, in bytes, of the original input file that RICOH ProcessDirector used to create the job.		No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Instance	General: Parent server	Contains the name of the RICOH ProcessDirector server that owns the submit step for the input file.		No	No
Job.JDF.JobID	General: JDF job ID	Shows the value of the job ID in the JDF job ticket received by RICOH ProcessDirector with the job.		No	Yes
Job.JDF. JobPartID	General: JDF part ID	Shows the value of the part ID for the job in the JDF job ticket received by RICOH ProcessDirector with the job.		No	Yes
Job.JobSize	Status: Job size (sheets)	Contains a calculated value that RICOH ProcessDirector uses when it schedules jobs to printers. It also uses the value of this property when a workflow uses the VerifyPrinted- SheetCount step to validate the actual number of sheets that print for a job.		No	No
Job.JobType	General: Workflow	Contains the name of the workflow that defines the processing phases and steps for the job.		No	No
Job.JobType. History	Status: Workflow history	Contains a list of workflows that define processing phases and steps that the job has passed through and where you can restart the job.		No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Locations	Scheduling: Requested location	Specifies the location where the job should print.		Yes	No
Job.Media	Scheduling: Media (ready supported all)	Specifies the media to use for the job.		Yes	Yes
Job. MediaRequired	Media required	Shows the names of the media objects specified for the entire job or page exception. You cannot change the value of this property.		No	Yes
Job.Name	General: Job name	Contains the name of the job.		Yes	Yes
Job.OutputBin	Scheduling: Output bin (requested available all)	Specifies the output bin to use for the job.		Yes	Yes
Job. OutputFormat	Scheduling: Output format	Used to specify whether the first page or the last page of a job prints first. The value can also indicate whether the job needs to be transformed to a different data stream before it is printed.	<ul style="list-style-type: none"> • AtoZforAFP • AtoZforPDF • ZtoAforAFP • ZtoAforPDF • Transform 	No	No
Job.PageLength	Transform: Transform page length (<i>unit</i>)	Specifies the length in inches or millimeters of the image that the Transform program generates.		Yes	No
Job.PageRange	Print: Pages to print again	Specifies which pages in the current job are printed again.		No	No
Job.PageWidth	Transform: Transform page width (<i>unit</i>)	Specifies the width in inches or millimeters of the image that the		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		Transform program generates.			
Job. PagesStacked	Status: Pages stacked	Contains the number of logical pages that have printed and that have reached the output stacker of the printer device.		No	No
Job.PDF. ActionList	Enhance PDF: Action list	Specifies one or more actions that a step based on the EnhancePDF step template uses to manipulate or evaluate a PDF file.		Yes	No
Job. PDFCheckResult	Enhance PDF: PDF results check	Shows whether the content of a PDF file meets PDF specifications. To check the content, you add a step based on the EnhancePDF step template to your workflow. In the step, specify the CheckPDF action with the -RPDproperty set to Job.PDFCheckResult.		No	No
Job.PDF. FinisherOrder- Configuration	PDF: Finisher order	For 2-up printing, specifies whether to place the content on the left page first and then the right, or on the right page first and then the left.	<ul style="list-style-type: none"> • LeftToRight • RightToLeft 	Yes	No
Job.PDF. NUpConfigura- tion	PDF: N-Up	Specifies how many pages print side by side on the same sheet.	<ul style="list-style-type: none"> • 1 • 2 	Yes	No
Job.PDF. Orientation	PDF orientation	Specifies the orientation to be used to print the job.	<ul style="list-style-type: none"> • Not set (default) • Portrait • Landscape 	No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.PDF. PageRotationFromOriginal	PDF: Additional page rotation	Specifies whether to change the print orientation of the pages in the job beyond any rotation added by the Leading edge into finisher property.	<ul style="list-style-type: none"> • 0 • 90 • 180 • 270 	Yes	No
Job.PDF. RollConfiguration	PDF: Leading edge into finisher	Specifies which edge of the print job enters the finisher first.	<ul style="list-style-type: none"> • JobEndEdgeIntoFinisher • JobStartEdgeIntoFinisher 	Yes	No
Job.Phase	Status: Current phase	Contains the name of the RICOH ProcessDirector phase that is currently processing the job.	<ul style="list-style-type: none"> • Complete • Prepare • Print • Receive 	No	No
Job. PhaseProgress	Status: Progress within the current phase	Contains the progress status for the job within the phase that the Job phase property identifies.	<ul style="list-style-type: none"> • Error • Manual • Staging • Working 	No	No
Job.Preview. AcceptedBy	Preview Print: Accepted by	Shows the user ID of the user who accepted the preview print.		No	No
Job.Preview. AutoAccept	Preview Print: Accept preview print automatically	Specifies whether RICOH ProcessDirector accepts the preview print automatically and moves the job to the next step in the workflow.	<ul style="list-style-type: none"> • No • Yes 	Yes	No
Job.Preview. PageRange	Preview Print: Page range for preview print	Shows a numeric string that describes which pages in the job are printed as samples in the PreviewPrint step.		Yes	No
Job.Preview. Requested Printer	Preview Print: Requested printer for preview print	Specifies the name of the printer that the PreviewPrint step		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		sends the preview print job to.			
Job.Print. AssignPrintTime	Status: Assigned to printer	Specifies the date and time when the printer received the job.	Dates and times are stored as Universal Time Code (UTC) values in this timestamp format: yyyy-mm-dd hh:mm:ss.fffffffff <ul style="list-style-type: none"> • yyyy is the 4-digit year. • mm is the 2-digit numerical abbreviation for the month. • dd is the 2-digit day. • hh is the 2-digit hour. • mm is the 2-digit minute. • ss is the 2-digit second. • fffffffff is the fraction of a second to 9 decimal places and is optional. 	No	No
Job.Print. CumulativePagesStacked	Status: Cumulative pages stacked	Contains the total number of logical pages that have printed and reached the output stacker of the printer, including any reprinted pages.		No	No
Job.Print. Cumulative-SheetsStacked	Status: Cumulative sheets stacked	Contains the total number of physical sheets that have printed and reached the output stacker of the printer throughout the life of the job on the RICOH ProcessDirector system.		No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Print. EndPrintTime	Status: Time printing completed	Shows the date and time when the printer finished printing the job successfully.	Dates and times are stored as Universal Time Code (UTC) values in this timestamp format: yyyy-mm-dd hh:mm: ss.fffffffff <ul style="list-style-type: none"> • yyyy is the 4- digit year. • mm is the 2-digit numerical abbreviation for the month. • dd is the 2-digit day. • hh is the 2-digit hour. • mm is the 2-digit minute. • ss is the 2-digit second. • fffffffff is the fraction of a second to 9 decimal places and is optional. 	No	No
Job.Print. HeaderConfig	Banner Pages: Header page configuration file	Specifies the path to and the name of the configuration file that RICOH ProcessDirector uses to create the content and format of the header page for the job.		Yes	No
Job.Print. HeaderCopies	Banner Pages: Header copies	Contains the number of copies of the job header page that RICOH ProcessDirector should print before it begins to print the job itself.		Yes	No
Job.Print.SpoolID	General: Spool ID	Identifies the spool ID that RICOH ProcessDirector generates for the job		No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
		before it sends the job to its printer driver component.			
Job.Print.TrailerConfig	Banner Pages: Trailer page configuration file	Specifies the path to and the name of the configuration file that RICOH ProcessDirector uses to create the content and format of the trailer page for the job.		Yes	No
Job.Print.TrailerCopies	Banner Pages: Trailer copies	Contains the number of copies of the job trailer page that RICOH ProcessDirector should print after the job prints.		Yes	No
Job.Priority	Scheduling: Job priority	Contains the printing priority of the job.		Yes	Yes
Job.ProcessGroupID	General: Process group ID	Specifies the number of the processing group, if any, to which the job belongs.		No	No
Job.ProcessGroupOrder	General: Process group order	Identifies the position of the job, if any, in a group of jobs.		No	No
Job.Punch	Scheduling: Punch	Specifies the number and position of holes to punch in the output.	<ul style="list-style-type: none"> • 2_at_bottom • 2_at_left • 2_at_right • 2_at_top • 3_at_bottom • 3_at_left • 3_at_right • 3_at_top • 4_at_bottom • 4_at_left • 4_at_right • 4_at_top • Multiple_at_bottom • Multiple_at_left • Multiple_at_left 	Yes	Yes

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
			<ul style="list-style-type: none"> Multiple_at_top 		
Job.ReprintCount	Status: Reprint count	Shows how many times a job has been reprocessed for printing.		No	No
Job. RequestedPrinter	Scheduling: Requested printer	Contains the name of the printer that was requested for the job.		Yes	Yes
Job.Resolution	Transform: Transform resolution (dpi)	For the standard transform features, specifies the resolution of the full page of image output that the data transform program generates. Specify a value that is appropriate for the model of the printer that prints the job.		Yes	No
Job.RestartSteps	Displays as the Phase and step list on the Process job again page	Shows the phases and associated steps that authorized users can select to start processing the job again.		No	No
Job. RetainDuration	General: Retention period (<i>unit</i>)	Controls the amount of time in minutes, hours, or days that RICOH ProcessDirector retains a job after it reaches the RetainCompleted- Jobs step in the Complete phase.		Yes	Yes
Job. RetainStartTime	General: Retention start time	Contains the time at which the retention period for a job in the Complete phase began.	Dates and times are stored as Universal Time Code (UTC) values in this timestamp format: yyyy-mm-dd hh:mm: ss.ffffffffff	No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
			<ul style="list-style-type: none"> • yyyy is the 4-digit year. • mm is the 2-digit numerical abbreviation for the month. • dd is the 2-digit day. • hh is the 2-digit hour. • mm is the 2-digit minute. • ss is the 2-digit second. • fffffff is the fraction of a second to 9 decimal places and is optional. 		
Job. SavedSearches	Saved Filters	Lets you select from the filters you previously saved to the Job table.		No	No
Job. SetJobOverrides	Job values file	Specifies the full path and file name of a configuration file that is used to set property values.		Yes	No
Job. SetJobOverrides	Job values file	Specifies the full path and file name of a configuration file that is used to set property values.		Yes	No
Job. SheetsStacked	Status: Sheets stacked	Shows the number of physical sheets that have printed and that have reached the output stacker of the printer device.		No	No
Job. SnapshotJobFile. FileToBeCopied	File type to be copied	Specifies the usagetype and datatype of the file this step should find in the spool directory and copy for later use.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job. SnapshotJobFile. NewFileDescriptor	Snapshot file descriptor	The text that the step inserts in the file name between the job ID and the datatype when it saves the snapshot of the job file.		Yes	No
Job. SourceInputDeviceName	General: Source input device	Shows the name of the input device for the job on the system where the job originated.		No	No
Job. SpoolFileStem	General: Root file path	Specifies the directory location for all files associated with this job.		No	No
Job.Staple	Scheduling: Staple	Specifies the number and position of staples to use in the output.	<ul style="list-style-type: none"> • 2_at_bottom • 2_at_center • 2_at_left • 2_at_right • 2_at_top • Bottom_left • Bottom_right • Top_left • Top_left_diagonal • Top_left_horizontal • Top_left_vertical • Top_right • Top_right_diagonal • Top_right_horizontal • Top_right_vertical 	Yes	Yes
Job. StapleRequired	Stapling required	Shows whether this job or any of its page exceptions must be stapled. You cannot change the value of this property.		No	Yes
Job.State	Status: Current job state	Contains the current processing state of the job.	<ul style="list-style-type: none"> • Assigned • Complete • Creating 	No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
			<ul style="list-style-type: none"> • Error • ManualWaiting • ManualWorking • Printing • Processing • Queued • Release • Retained • Spooling • Stopped • Unassigned • Waiting 		
Job.Step	Status: Current step	Contains the name of the step that is processing the job.		No	No
Job.StopAtPhase	Status: Stop when entering phase	Specifies whether RICOH ProcessDirector stops a job when it enters the first step of a specific phase.	<ul style="list-style-type: none"> • Complete • Prepare • Print • Receive 	Yes	No
Job.SubmitTime	Scheduling: Time submitted	Contains the date and time when the input device submitted the input file and created the corresponding RICOH ProcessDirector job.	<p>Dates and times are stored as Universal Time Code (UTC) values in this timestamp format: <code>yyyy-mm-dd hh:mm:ss.fffffff</code></p> <ul style="list-style-type: none"> • yyyy is the 4-digit year. • mm is the 2-digit numerical abbreviation for the month. • dd is the 2-digit day. • hh is the 2-digit hour. • mm is the 2-digit minute. • ss is the 2-digit second. • fffffff is the fraction of a second to 9 decimal places and is optional. 	No	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.TestJob	General: Test job	Specifies whether the job is a test job or a production job.	<ul style="list-style-type: none"> No Yes 	Yes	No
Job.TotalPages	Status: Total pages	Contains the total number of logical pages in the job.		No	No
Job.TotalSheets	Status: Total sheets	Contains the total number of physical sheets in the job.		No	No
Job.TrailerConfig	Trailer page configuration file	Specifies the path and name of the configuration file that RICOH ProcessDirector uses to create the content and format of the trailer page placed after each file that is contained in the ZIP file. Authorized users can specify one of the configuration files for trailer pages that RICOH ProcessDirector provides or they can specify a customized configuration file.		Yes	No
Job.UseProxy	Use proxy server	Specifies whether the DownloadFile step uses a proxy server to resolve the URL of the file to be downloaded.	<ul style="list-style-type: none"> No Yes 	Yes	No
Job.Validate. FileTo	Check File Structure: File to check	Specifies the full path and name of the JSON or XML file that a step based on the CheckFileStructure step validates by checking the file syntax.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Edit- able	Job ticket
Job.Validate. FileType	Check File Structure: File format	Specifies the format of the file that a step based on the CheckFileStructure step template validates.	<ul style="list-style-type: none"> • XML • JSON 	Yes	No
Job.Wait.Amount	Wait: Wait for	Specifies the amount of time the job waits before moving to the next step.		Yes	No
Job.Wait. TimeOfDay	Wait: Wait until	Specifies the specific time when the job moves to the next step.		Yes	No
Job.Wait. TimeZone	Wait: Time zone	Specifies the time zone to use with the Wait until property.		Yes	No
Job.Wait. WaitUntil	Wait: Wait step ends	Shows the date and time when the waiting period ends and the job moves to the next step.		Yes	No
Job.Wait. WhenToMove	Wait: Complete step after	When values are set for both the Wait until and the Wait for properties, this property specifies whether the Wait step completes when the earlier or later of the two times is reached.	<ul style="list-style-type: none"> • First occurs • Last occurs 	Yes	No
Job.WaitReason	Status: Reason for wait status	Identifies the condition that is preventing RICOH ProcessDirector from further processing of the job.	<ul style="list-style-type: none"> • DeviceUnavaila- ble • NoMatchingDe- vice • ServerUnavaila- ble • StepTemplate- Disabled 	No	No
Job.XML. JSONInputFile	XML: JSON input file	Specifies the JSON file to be converted into XML format.		Yes	No

Database name	Notebook tab: field name	Brief description	Internal values	Editable	Job ticket
Job.XML. XMLOutputFile	XML: XML output file	Specifies the location and name of the XML file created by the step.		Yes	No
Job.ZipFile. FilesToZip	ZIP Files: Files to ZIP	Specifies a comma-separated list of files that a step based on the ZIPFiles step template copies to create a ZIP file.		Yes	No
Job.ZipFile. ZipToSave	ZIP Files: Output file	Specifies the full path and name of the output file created by a step based on the ZIPFiles step template.		Yes	No
Job. ZipFilesToEmail	Email: Attach ZIP file	Specifies whether the file or files should be packaged as a ZIP file before they are attached to the email.	<ul style="list-style-type: none"> • No • Yes 	Yes	No

Database property names for workflows

Messages about jobs might refer to workflow properties by their database names, which begin with **JobType**.

In the **Editable** column:

- Yes means that an authorized user can change the value after the workflow has been created.
- No means that an authorized user cannot change the value.

Workflow properties

Database name	Notebook field name	Brief description	Editable
Connector. AnyOrAllFieldHelp	Apply any or all of the following conditions	Specifies whether a job must meet all of the conditions specified by a rule or any one of the conditions.	Yes
Connector. ConditionPredicate	Summary	The Summary field shows your selections in the Conditions area in a database query format. The value changes as you modify conditions.	Yes

Database name	Notebook field name	Brief description	Ed-itable
Connector.JsonRule	Conditions	Lets you define one or more conditions for a rule. Each condition consists of a job property, a comparison, and a value.	Yes
Connector.Order	Order of execution	When multiple connectors exit a single step, RICOH ProcessDirector must determine which connector the job should follow to the next step. This value indicates which connector RICOH ProcessDirector should evaluate first, second, third, and so on, when it tries to choose which connector the job should use.	Yes
Job.CurrentDay	Current day	Specifies the current day of the week of the server RICOH ProcessDirector is running on when a condition requesting it is evaluated.	Yes
Job.CurrentTime	Current time	Specifies the current time of the server that RICOH ProcessDirector is running on when a condition requesting it is evaluated.	Yes
JobType.Changeld	Alternate ID	Specifies an alternate ID for the workflow.	Yes
JobType.CreatedBy	Created by	Specifies the ID of the extension that created the workflow.	No
JobType.Description	Description	Contains text that describes the workflow.	Yes
JobType.GroupName	Group name	Specifies the name of the group the workflow belongs to.	Yes
JobType.LastModified	Last modified	The date and time that the workflow was last changed.	No
JobType.Location	Workflow location	Specifies the location associated with a workflow.	Yes
JobType.ModifiedBy	Modified by user	Specifies the user name of the user who made the last change to this workflow.	No
JobType.Owner	Owner	Specifies the owner of the workflow.	Yes
JobType.SourceID	Source ID	Specifies the ID of the workflow.	No

Database name	Notebook field name	Brief description	Ed-itable
StepChain.Color	Color	Specifies the color of the step chain as it appears in the workflow.	Yes
StepChain.Description	Step chain description	Describes the function of the step chain.	Yes
StepChain.ID	Step chain name	Specifies the name for the step chain.	No
StepChain.LastModified	Last modified	The date and time that the step chain was last changed.	No
StepChain.ModifiedBy	Modified by user	Specifies the name of the user who made the last change to this step chain.	No
StepChain.Owner	Owner	Specifies the owner of the step chain.	Yes
StepChain.Usage	Step chain usage	Specifies what the group of steps in the step chain are used for.	Yes

Database property names for printers

Messages about printers might refer to properties by their database names. Not all properties are applicable to all types of printers.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that an authorized user can change the value after the printer has been created.
- No means that an authorized user cannot change the value.

Printer properties

Database name	Notebook tab: field name	Brief description	Internal values	Editable
CustomPDFPrinter.ImportPrinter	General: Custom PDF Printer Server	Select the printer server where you want to define a Custom PDF printer device.		No
CustomPDFPrinter.PrinterType	General: Custom printer type	Specifies the type of printer to define as a Custom PDF printer object.		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
JdfDirectPrinter. VPNName	General: Virtual printer name	Specifies the name of the virtual printer as defined on the Ricoh TotalFlow printer object.		Yes
JdfOutputPrinter. PrinterType	General: Type of printer	Specifies the printer type.	The list of all printers that can be connected as Ricoh PDF printers.	Yes
JdfOutputPrinter. BannerPage.Tray	Banner Pages: Banner page input tray	Specifies the printer's input tray that holds the paper for header and trailer pages.		Yes
JdfOutputPrinter. JobStatusPolling	General: Job status polling interval	Defines how frequently the Ricoh PDF or Custom PDF printer requests updates about the status of a job.		Yes
JdfOutputPrinter. MergeBanner	Banner Pages: Merge banner pages into PDF print file	Specifies whether or not header and trailer banner files are merged into the PDF print file or sent to the printer as separate files.	<ul style="list-style-type: none"> • No • Yes 	Yes
JdfOutputPrinter.Port	General: Printer port	Specifies the communications port for the printer.		Yes
JdfOutputPrinter.URL	General: Printer URL	Specifies the host name or TCP/IP address that is used to communicate with the printer.		Yes
PassThroughPrinter. CodePage	General: Code page	Specifies the code page that the printer uses to interpret the printer command.		Yes
PassThroughPrinter. Command	General: Printer command	The command that is used to submit jobs to the printer.		Yes
PassThroughPrinter. ControlFileTemplate	General: Control file template	Specifies the full path name of the control		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		file template that maps job properties to printer command parameters.		
PassThroughPrinter. MergeBanner	Banner Pages: Merge banner pages into PDF print file	Specifies whether or not header and trailer banner files are merged into the PDF print file or sent to the printer as separate files.	<ul style="list-style-type: none"> • No • Yes 	Yes
PassThroughPrinter. ValidRCs	General: Valid return codes	Contains a list of return code values that the printer command can issue to indicate that the job printed successfully.		Yes
Printer. CommunityName	SNMP: Community name	Specifies the name of the Simple Network Management Protocol (SNMP) community to which the printer belongs.		Yes
Printer.CurrentJobID	Status: Current job number	Specifies the job number of the job that is currently assigned to the printer.		No
Printer. CurrentJobName	Status: Current job name	Specifies the name of the job that is currently assigned to the printer.		No
Printer. CurrentJobPages-Stacked	Status: Current job pages printed	Shows the number of pages that have been printed for the job that is currently assigned to the printer.		No
Printer. CurrentJobProgress	Status: % Printed	Shows the current page that is printing for the job that is currently assigned to the printer, as a		No

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		percentage of the total pages in the job.		
Printer. CurrentJobTotalPages	Status: Total pages in current job	Shows the total number of pages in the job that is currently assigned to the printer.		No
Printer.CurrentStatus	Status: Last status message	If the printer status is Disconnected or Needs attention , contains the most recent error or warning message received about the printer. Otherwise, contains the most recent informational message received about the printer.		No
Printer.CustomerName	Scheduling: Customer name	Specifies the customer name that is assigned to the printer.		Yes
Printer.Description	General: Printer description	Contains text that describes the printer object.		Yes
Printer.Enabled	Status: Enabled status	Specifies whether RICOH ProcessDirector can schedule jobs to the printer.		No
Printer. EnhancePDFFilter	General: Action list	Specifies one or more actions that can be applied to a PDF file and JDF file before it is sent to the Ricoh PDF or Ricoh TotalFlow printer.		Yes
Printer. FirstSegmentSize	General: PDF first segment size	Specifies the size, in pages, of the first PDF job segment that is sent to the printer.		No

Database name	Notebook tab: field name	Brief description	Internal values	Editable
Printer.FoldCapable	Scheduling: Folding capable	Specifies whether the printer can fold pages in different ways depending on the job assigned.		Yes
Printer.HeaderExit	Banner Pages: Enable header pages	Controls whether the printer prints a header page at the beginning of each job.	<ul style="list-style-type: none"> • No • Yes 	Yes
Printer.ID	Displayed in the properties notebook title: Printer name	Contains the name of the printer.		No
Printer.Instance	General: Printer server	Specifies the name of the RICOH ProcessDirector server that sends jobs to this printer.		Yes
Printer.JobSize	Scheduling: Job size supported	Specifies the size of the jobs in sheets that RICOH ProcessDirector can schedule to the printer.		Yes
Printer.Language	General: Printer language	Indicates what language the printer driver component uses when it returns messages to RICOH ProcessDirector.		Yes
Printer.LastModified	General: Last modified	The date and time that the printer was last changed.		No
Printer.Locations	Scheduling: Printer location	Specifies the location of the printer.		Yes
Printer. MaxConcurrentJobs	General: Maximum concurrent jobs	Specifies the maximum number of jobs that the printer driver component of RICOH ProcessDirector can control at the same time.		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
Printer.Media	Scheduling: Media supported	Specifies the media that the printer supports.		Yes
Printer.MediaCatalog	Media: Media to use	Specifies whether the media information sent to the printer for a job is system media or printer media.	<ul style="list-style-type: none"> • Printer • System 	Yes
Printer.Model	General: Printer paper type	Contains the type of the printer; for example, continuous-form or cut sheet.		Yes
Printer.Model.Specific	General: Printer model	Contains the model number of the printer.		Yes
Printer.ModifiedBy	General: Modified by user	Specifies the user name of the user who made the last change to this printer.		No
Printer.OutputBin	Scheduling: Output bins available	Lists the output bins that can be installed on the printer.		Yes
Printer.OutputFormat	Scheduling: Output format	Specifies whether the printer is set up to print the first page or the last page of a job first, usually based on finishing or other post-processing requirements. The value can also indicate that the job must be transformed to a different datastream before it can be printed.	<ul style="list-style-type: none"> • AtoZforAFP • AtoZforPDF • ZtoAforAFP • ZtoAforPDF • Transform 	Yes
Printer.PerfectBindingCapable	Scheduling: Perfect binding capable	Specifies whether the printer can glue a cover onto the binding edge of paper.		Yes
Printer.PunchCapable	Scheduling: Punch capable	Specifies whether a finisher attached to		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		the printer can punch holes in the output.		
Printer.Retry	Connection: Printer connection retry count	Specifies how many times RICOH ProcessDirector tries to connect to the printer if an earlier attempt failed.		Yes
Printer.RetryInterval	Connection: Retry interval	Specifies how often RICOH ProcessDirector tries to connect to the printer if an earlier attempt failed.		Yes
Printer. RingBindingCapable	Scheduling: Ring binding capable	Specifies whether the printer can insert rings along the binding edge of paper.		Yes
Printer.S2VBarcode	AFP: Send blank pages after job	Specifies the number of blank pages to send to the printer after the last job that is queued to the printer prints.		Yes
Printer.SegmentSize	General: PDF segment size	Specifies the size, in pages, of PDF job segments that are sent to the printer.		Yes
Printer.SerialNumber	Status: Serial number	Contains the serial number of the printer.		No
Printer.SNMPStatus	Status: SNMP status	Indicates whether SNMP is connected, disconnected, or disabled.		No
Printer.StapleCapable	Scheduling: Staple capable	Specifies whether a finisher attached to the printer can staple the output.		Yes
Printer.Status	Status: Printer status	Contains the current status of the printer.		No

Database name	Notebook tab: field name	Brief description	Internal values	Editable
Printer.TCPIP.Address	General: Printer TCP/IP address or host name	Specifies either the network TCP/IP address or the fully qualified host name of the printer hardware.		Yes
Printer.TrailerExit	Banner Pages: Enable trailer pages	Specifies whether the printer prints a trailer page at the end of each job.	<ul style="list-style-type: none"> • No • Yes 	Yes
Printer.UseSNMP	SNMP: Use SNMP	Specifies whether RICOH ProcessDirector uses SNMP to monitor the printer.		Yes
Printer.UseSnmpUpdateMedia	SNMP: Get tray information from printer	Specifies whether RICOH ProcessDirector uses the Simple Network Management Protocol (SNMP) to update media.	<ul style="list-style-type: none"> • No • Yes 	Yes
Printer.Version	Status: Version	Contains the printer version returned by SNMP.		No

Database property names for media

Messages about media might refer to properties by their database names, which begin with **Media**.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:


- **Yes** means that an authorized user can change the value after the media has been created.
- **No** means that an authorized user cannot change the value.

In the **Job ticket** column:

- **Yes** means that the property is compared to one or more values in the job ticket that is used to submit the job in order to set the **Media** property for the job.
- **No** means that the property is not compared to values in the job ticket.

Media properties

Database name	Field name	Brief description	Internal values	Editable	Job ticket
Media.Description	Media description	Contains user-defined text that describes the media.		Yes	No
Media.ID	Displayed in the properties notebook title: Media name	Contains the name of the media.		No	Yes
Media.LastModified	Last modified	The date and time that the media was last changed.		No	No
Media.MappedMedia	Mapped system media	Specifies the system media that you want to map to this printer media.		Yes	Yes
Media.MappedMedia.PrinterFilter	Printer	Specifies what printers to display media mappings for in the table.	<ul style="list-style-type: none"> All 	Yes	Yes
Media.ModifiedBy	Modified by user	Specifies the user name of the user who made the last change to this media.		No	No
Media.Printer	Printer	Specifies the name of the printer that this printer media is associated with.		No	Yes
Media.ProductId	Product ID	Specifies the product ID of the media. This property is used only with jobs sent to Ricoh PDF printers with a Data stream to send value of JDF/PDF .		Yes	Yes
Media.SendName	Send media name in job ticket	Specifies whether the media name is sent to the Ricoh PDF printer in the JDF job ticket.		Yes	No
MediaSize.Height	Media height	Specifies the height of the media.		Yes	Yes

Database name	Field name	Brief description	Internal values	Editable	Job ticket
MediaSize. Units	Media units	Specifies the unit of measure for the media height and width.	<ul style="list-style-type: none"> • Inches • Millimeters • Points 	Yes	No  Note Media dimensions in the job ticket are always in points.
MediaSize. Width	Media width	Specifies the width of the media.		Yes	Yes
MediaType. Color	Media color	Specifies the color of the media.	<ul style="list-style-type: none"> • Blue • Buff • Goldenrod • Gray • Green • Ivory • None • Orange • Pink • Purple • Red • User • White • Yellow 	Yes	Yes
MediaType. Details	Media details	Specifies the general category to which the media belongs; for example, letterhead or transparency.	<ul style="list-style-type: none"> • Bond • Cardstock • Envelope • Labels • Letterhead • Paper • Special • Tabstock • Translucent • Transparent 	Yes	Yes
MediaType. Preprinted	Media is preprinted	Specifies whether the media is preprinted.	<ul style="list-style-type: none"> • No • Yes 	Yes	Yes
MediaType. Punched	Media is prepunched	Specifies whether the media is prepunched.	<ul style="list-style-type: none"> • No • Yes 	Yes	Yes
MediaType. Recycled	Media is recycled	Specifies whether the media is recycled.	<ul style="list-style-type: none"> • No • Yes 	Yes	Yes
MediaType. Weight	Media (gsm)	Specifies the weight of the media in grams per square meter (gsm).		Yes	Yes

Database property names for input devices

Messages about input devices might refer to properties by their database names. Some properties are specific to Hot folder input devices; their database property names begin with **HotFolder**. Others are specific to Download input devices; their database property names begin with **zOSDownload**.

Properties whose names begin with **InputDevice** apply to all types of input devices.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that an authorized user can change the value after the input device has been created.
- No means that an authorized user cannot change the value.

Input device properties

Database name	Notebook tab: field name	Brief description	Internal values	Editable
HotFolder. CodePage	General: Device code page	The code page that the input device uses to read the contents of any files that accompany the print files, such as trigger files and list files.	<ul style="list-style-type: none"> • euc_jp • iso8859_1 • iso8859_15 • utf8 	Yes
HotFolder. SetPattern	Batching: Matching pattern for sets	Specifies the file name pattern that an input device uses when it creates sets. This pattern is a regular expression that indicates the portion of the input file names that must match within a set. The input files that make up a set are determined by the values for the Data patterns, JDF patterns, Overrides patterns, and File Patterns properties.		Yes
InputDevice. BatchingMethod	Batching: Batching method	Specifies how the input device groups multiple input files or sets of files and submits them as a single job or parent and child jobs. You can change this value for Hot folder input	<ul style="list-style-type: none"> • AutoBatch • JDF • ListFile • MVSOut-Grp • None • Pattern-Based 	Yes (HotFolder) No (Download, LPD)

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		devices, but not for Download input devices or for LPD input devices.	<ul style="list-style-type: none"> • Number-OfFiles • Number-OfPages • Number-OfSets • PagesIn-Sets • SetsBy-Time • Time 	
InputDevice.Child. InitJobTypeStep	General: Child workflow initialization step	Specifies the name of the step that the input device uses to initialize the workflow for single jobs or child jobs, convert an optional overrides file submitted with a job to a text file in <i>property name=value</i> format for setting job properties, or both.		Yes
InputDevice.Child. JobType	General: Child workflow	Specifies the name of the workflow that the input device assigns to jobs that contain only a single file or jobs that are children of a parent job.		Yes
InputDevice.Child. JobType ParsingRules	Advanced: Child workflow parsing rules	Contains the path and file name of a control file.		Yes
InputDevice.Child. JobTypeFilename Pattern	Advanced: Child workflow pattern	Contains a pattern-matching string that indicates what part of the file name should be used for the workflow for single and child jobs.		Yes
InputDevice. ConvertOverrides	Advanced: Convert overrides	Specifies whether a control file is required to convert the overrides file submitted with a job to a job properties file in RICOH	<ul style="list-style-type: none"> • No • Yes 	Yes (HotFolder) No (Download, LPD)

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		ProcessDirector <i>property name=value</i> format.		
InputDevice. CreateZip	Batching: Create . zip file	Specifies whether the input device submits data files as individual jobs or collects them in a .zip file and submits them as a single job.	<ul style="list-style-type: none"> • No • Yes 	Yes
InputDevice. Description	General: Input device description	Contains text that describes the input device.		Yes
InputDevice. Enabled	Status: Enabled status	Specifies whether this input device can submit input files that it receives to create the corresponding RICOH ProcessDirector jobs.	<ul style="list-style-type: none"> • No • Yes 	No
InputDevice. FileCompletion	Advanced: Completion method	Specifies how the input device determines that file transmission is complete for an input file.	<ul style="list-style-type: none"> • CheckOpen • CheckSize • None • Trigger 	Yes (HotFolder) No (Download, LPD)
InputDevice. FileDataPatterns	Batching: Data patterns	Contains one or more pattern-matching strings that RICOH ProcessDirector uses to identify the input files that the input device should accept for processing as single jobs.		Yes (HotFolder) No (Download, LPD)
InputDevice. FileJDFPatterns	Batching: JDF patterns	Contains one or more pattern-matching strings that RICOH ProcessDirector uses to identify Job Definition Format (JDF) job tickets.		Yes (HotFolder) No (Download, LPD)
InputDevice. FileListPatterns	Batching: List patterns	Contains one or more pattern-matching strings that RICOH ProcessDirector uses to identify list files.		Yes (HotFolder) No (Download, LPD)
InputDevice. FileOtherPatterns	Batching: Overrides patterns	Contains one or more pattern-matching strings that RICOH		Yes (HotFolder) No

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		ProcessDirector uses to identify files that it must process with a list file and the input files that the list file specifies, or with another single input file.		(Download, LPD)
InputDevice. FileTrigger Patterns	Advanced: Trigger patterns	Contains one or more user-defined pattern-matching strings that RICOH ProcessDirector uses to identify trigger files.		Yes (HotFolder) No (Download, LPD)
InputDevice. FolderLocation	General: Folder location	Contains the name of the directory that the input device monitors for incoming jobs.		Yes
InputDevice. Frequency	Batching: Batching interval	Indicates the time interval used to submit a batch of input files.		Yes
InputDevice.ID	Displayed in the property notebook title	Contains the name of the input device.		No
InputDevice. InitJobTypeStep	General: Workflow initialization step	Specifies the name of the step that the input device uses to initialize the parent workflow for the input files that the input device receives, convert an optional overrides file submitted with a job to a text file in RICOH ProcessDirector <i>property name=value</i> format for setting job properties, or both.		Yes
InputDevice. Instance	General: Parent server	Specifies the name of the RICOH ProcessDirector server that receives and records messages for this input device.		Yes
InputDevice. JobType	General: Workflow	Specifies the name of the workflow that the		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		input device assigns to the job.		
InputDevice. JobTypeFilename Pattern	Advanced: Parent workflow pattern	Contains a pattern- matching string that indicates what part of the file name should be used for the workflow for the parent job.		Yes
InputDevice. JobTypeParsing Rules	Advanced: Parent workflow parsing rules	Contains a path and file name of a control file.		Yes
InputDevice. LastModified	General: Last modified	The date and time that the input device was last changed.		No
InputDevice. Locations	General: Input device location	Contains the location associated with the input device.		Yes
InputDevice. MaxErrors	General: Maximum errors	Contains the number of communication errors that can occur for the input device before RICOH ProcessDirector disables the input device.		Yes
InputDevice. ModifiedBy	General: Modified by user	Specifies the user name of the user who made the last change to this input device.		No
InputDevice. NumberOfFiles	Batching: Number of files to batch	Specifies the number of files that are combined in a single submission when you choose the Number batching method in the General tab.		Yes
InputDevice. NumberOfPages	Batching: Number of pages to batch	Specifies the maximum number of PDF pages that should be combined in a single submission when you choose the Pages batching method in the General tab.		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
InputDevice. PageThreshold	Batching: Exceed pages to batch	Specifies whether the Hot folder should include the file that exceeds the value for the Number of pages to batch property when it submits a collection of PDF files.		Yes
InputDevice. PollInterval	General: Polling interval (<i>unit</i>)	Specifies the time interval at which RICOH ProcessDirector checks for files in the directory that the Folder location property of the Hot folder input device specified.		Yes
InputDevice. ScheduleDaily	Batching: Frequency (days)	Specifies how often the input device submits a batch of input files.		Yes
InputDevice. ScheduleHourly	Batching: Frequency (hours)	Specifies how often the input device submits a batch of input files.		Yes
InputDevice. ScheduleMinute	Batching: Frequency (minutes)	Specifies how often the input device submits a batch of input files.		Yes
InputDevice. ScheduleMonthly	Batching: Frequency (months)	Specifies how often the input device submits a batch of input files.		Yes
InputDevice. ScheduleWeekly	Batching: Frequency (weeks)	Specifies how often the input device submits a batch of input files.		Yes
InputDevice. StagingLocation	General: Staging location	Contains the name of the directory into which the input device moves the input file before submitting it as a job.		Yes
InputDevice. StartDate	Batching: Batching start date	Specifies the date when the Time batching method takes effect for the input device.		Yes
InputDevice. StartDateAndTime	Batching: Batching start date and time	Specifies the date and time when the Time or		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
		Sets by time batching method takes effect for the input device.		
InputDevice.Status	Status: Connection status	Lists the current status of the input device: connected, disconnected, or unable to connect.	<ul style="list-style-type: none"> • Connected • Disconnected • UnableToConnect 	No
InputDevice.SubmitStep	General: Submit step	Contains the name of the submit step to which the input device sends the input file for job processing.		Yes
InputDevice.WaitingFileCount	Status: Input files waiting	Contains the number of input files that the input device has received but has not yet submitted to create the corresponding RICOH ProcessDirector jobs.		No
InputFilePattern.FilePattern	Batching: File Pattern	Specifies one or more file pattern-matching strings.		Yes
InputFilePattern.Required	Batching: File pattern required	Specifies whether a file that matches the input file pattern must exist in the Hot folder before submitting the job.	<ul style="list-style-type: none"> • No • Yes 	Yes
InputFilePattern.Sequence	Batching: File pattern sequence	Specifies the priority of the file pattern. Patterns are checked in this order.		Yes
InputFilePattern.SpoolFileType	Batching: Spool file type	Specifies the content or the data stream type of the file.		Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
InputFilePattern. SpoolFileUsage	Batching: Spool file usage	Specifies the purpose or role of the file within the job.		Yes
LPD.Codepage	General: Device code page	Specifies the code page that the LPD input device uses to read the contents of any files that accompany the print files, such as control files and list files.	<ul style="list-style-type: none"> euc_jp iso8859_1 iso8859_15 utf8 	Yes

Database property names for input files

Messages about input files might refer to properties by their database names, which begin with **InputFile**.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that an authorized user can change the value after the input file has been created.
- No means that an authorized user cannot change the value.

Input file properties

Database name	Column heading: Input files table	Brief description	Internal values	Editable
InputFile.JobType	Workflow	Contains the name of the workflow that the input device assigns to the input file when it submits the file for job processing.		No
InputFile.Location	Input file	Specifies the path to and the name of the input file.		No
InputFile.Status	Status	Contains the current status of the input file.	<ul style="list-style-type: none"> Error Processing Queued Waiting 	No
InputFile.SubmitGroup	Group	Identifies a member of a set of input files that		No

Database name	Column heading: Input files table	Brief description	Internal values	Editable
		were submitted as a group through a list file.		
InputFile.SubmitType	Submit type	Contains the submission type for the input file.	<ul style="list-style-type: none">• AIWList• Data• List• Other• Trigger	

Database name	Column heading: Input files table	Brief description	Internal values	Editable
InputFile.Timestamp	Rec'd	Contains the date and time when the input file arrived at the input device.	Dates and times are stored as Coordinated Universal Time (UTC) values in this timestamp format: yyyy-mm-dd hh:mm:ss. ffffffff <ul style="list-style-type: none"> • yyyy is the 4-digit year. • mm is the 2-digit numerical abbreviation for the month. • dd is the 2-digit day. • hh is the 2-digit hour. • mm is the 2-digit minute. • ss is the 2-digit second. • fffffff is the fraction of a second to 9 decimal places and is optional. 	No

Database property names for notification objects

Messages might refer to Notification properties.

In the **Editable** column:

- Yes means that an authorized user can change the value.
- No means that an authorized user cannot change the value.

Workflow properties

Database name	Notebook field name	Brief description	Editable
User. EmailAddress	Email address	Specifies the email address for the user.	Yes
Notification. AnyAllCustom	Apply any or all of the following conditions	Specifies how the conditions for a notification object are combined.	Yes
Notification. BCC	Blind copy address	One or more email addresses to send a blind copy of the notification email to. This value is used in the BCC: field of the email.	Yes
Notification.CC	Copy address	One or more email addresses to send a copy of the notification email to. This value is used in the CC: field of the email.	Yes
Notification. Condition	Conditions	Lets you define one or more conditions for a notification object. Conditions limit the notifications that are sent for the specified event. Each condition consists of an object property, a comparison, and a value.	Yes
Notification. Condition. NotifyWhen	Notify when	Specifies how the conditions for a notification object are combined.	Yes
Notification. Description	Notification description	The description might indicate when a notification is sent. For example, the description might say: "Email second shift operators if any printer is disabled after midnight."	Yes
Notification. EmailMessage	Message	Specifies the message to include in the body of the notification email. Messages can consist of plain text, HTML, and embedded images.	Yes
Notification. EmailSubject	Subject line	Specifies the text to include in the subject line of the notification email.	Yes
Notification. Enabled. Description	Enabled status	Specifies whether this notification object can send a notification.	No
Notification. Event	Event	Lets you define one or more properties to monitor for a notification object. Each event consists of an object property, an action, and optionally a value. The type of object whose properties are shown is set in the Event	Yes

Database name	Notebook field name	Brief description	Editable
		type property on the General tab of the Notification property notebook.	
Notification.EventType	Event type	Specifies the type of object to be monitored for notification. You can only use one type of Event type for each notification object.	Yes
Notification.ID	Notification name	Specifies the name of the notification object.	No
Notification.JsonCondition	Conditions	Lets you define one or more conditions for a notification object. Conditions limit the notifications that are generated for the specified event.	No
Notification.SecureConnection	Secure connection	Specifies whether the connection with the mail server should use SSL or TLS security for sending notification emails. The SMTP server must support SSL or TLS connections for this property to function.	Yes
Notification.SendLog	Attach log	Specifies whether the input device, job or printer log is attached to the notification email when a certain input device, job or printer event occurs, providing more information to the email recipients.	Yes
Notification.Threshold	Notification limit	Lets you specify how many notifications can be sent within a period of time. For example, if you specify 10 messages within 2 hours , a timer starts after the first notification is sent. If nine more notifications are sent in the next 30 minutes, no more notifications are sent until the timer reaches the 2 hour limit.	Yes
Notification.To	Recipient address	One or more email addresses to send the notification to. This value is used in the To: field of the email.	Yes
Notification.Type	Notification method	Specifies how notifications are delivered.	Yes
WorkflowSystem.EmailFrom	Sender email address	The email address used in the From: field for all notification emails.	Yes

Database property names for servers

Messages about servers might refer to properties by their database names, which begin with **Instance**.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that an authorized user can change the value after the server has been created.
- No means that an authorized user cannot change the value.

Server properties

Database name	Notebook field name	Brief description	Internal values	Editable
Instance.Description	Server description	Contains user-defined text that describes the RICOH ProcessDirector server.		Yes
Instance.Enabled	Enabled status	Specifies whether the server can do tasks.	<ul style="list-style-type: none"> • No • Yes 	No
Instance.ID	Displayed in the properties notebook title: Server name	Contains the name of the server.		No
Instance.IPAddress	Server IP address or host name	Specifies either the network IP address or the fully qualified host name of the computer that the server is running on.		Yes
Instance.InGeneralServerPool	In general server pool	Specifies whether the server is a general-usage server or a restricted-usage server.	<ul style="list-style-type: none"> • No • Yes 	Yes
Instance.LastModified	Last modified	The date and time that the server was last changed.		No
Instance.MaxHighUsageSteps	Maximum resource-intensive step count	Controls how many resource-intensive steps the server lets run concurrently.		Yes
Instance.MaxLowUsageSteps	Maximum step count for other steps	Controls how many non-resource-intensive steps the server lets run concurrently.		Yes
Instance.ModifiedBy	Modified by user	Specifies the user name of the user who made the last change to this server.		No
Instance.Status	Connection status	Shows the current status of the server.	<ul style="list-style-type: none"> • Connected • Disconnected 	No

Database property names for step templates

Messages about step templates might refer to properties by their database names, which begin with **StepTemplate**.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that an authorized user can change the value after the step template has been created.
- No means that an authorized user cannot change the value.

Step template properties

Database name	Notebook tab: field name	Brief description	Internal values	Editable
Step.Color	Step color	Specifies the color of the step as it appears in the workflow builder.	<ul style="list-style-type: none"> • Red • Orange • Gray • Blue • Purple • Not set 	Yes
Step.Description	Step description	Describes the purpose of the step. For example, it might describe how the step processes a job.		Yes
Step.DisplayName	General: Step name	The name given to a step when it is added to a workflow.		Yes
Step.ID	General: Step identifier	Specifies the internal name for this step. This value is derived from the name of the step.		No
Step.Template.CreatedBy	General: Template created by	Specifies the name of the feature or extension that created the step template.		No
StepTemplate.Color	Step color	Specifies the color of the step as it appears in the workflow builder.	<ul style="list-style-type: none"> • Red • Orange • Gray • Blue • Purple • Not set 	Yes

Database name	Notebook tab: field name	Brief description	Internal values	Editable
StepTemplate.Description	General: Template description	Contains text that describes the function of the step template.		Yes
StepTemplate.Enabled	General: Enabled status	Specifies whether the step template is enabled.	<ul style="list-style-type: none"> • No • Yes 	No
StepTemplate.HighResourceUsage	Tuning Properties page: Concurrent step limit	Specifies where the limits are set for the number of steps created from the step template that can run at the same time.	<ul style="list-style-type: none"> • Use limits set here • Use limits set on server 	Use limits set here
StepTemplate.ID	Displayed in the properties notebook title: Step template name	Contains the name of the step template.		No
StepTemplate.LastModified	Last modified	The date and time that the step was last changed.		No
StepTemplate.MaximumActiveCount	Tuning Properties page: Limit the number of concurrent steps active in the system to radio button	Specifies how many occurrences of the step template, and any steps that are created from it, can run concurrently on a specific type of RICOH ProcessDirector object.		Yes
StepTemplate.MaximumActiveUnit	Tuning Properties page: for each drop-down list	Specifies the type of RICOH ProcessDirector object to which the value of the Maximum active count property for the step template applies.	<ul style="list-style-type: none"> • Perl-nput-Device • Per-Printer • Per-Server • Per-System 	Yes
StepTemplate.ModifiedBy	Modified by user	Specifies the user name of the user who made the last change to this step.		No

Database name	Notebook tab: field name	Brief description	Internal values	Editable
StepTemplate. ModuleType	General: Module type	Identifies the type of function the step template provides.	<ul style="list-style-type: none"> • Clean-up • Init-Job-Type • Java • Manual • Print • Submit • Submit-Child 	No
StepTemplate.Servers	Tuning Properties page: Run only on the selected server or servers radio button	Lists all the restricted- usage servers and general-usage servers on which any steps that are created from the step template can run.		Yes
StepTemplate.SourceID	General: Step template source ID	Specifies the name of the step template that was used to create this step template.		No
StepTemplate. UseGeneralServerPool	Tuning Properties page: Servers to use	Specifies which computers can run the steps created by the step template.	<ul style="list-style-type: none"> • Run on specific servers • Run on servers in general server pool 	Run on servers in general server pool

Database property names for system properties

Messages about the RICOH ProcessDirector system might refer to properties by their database names, which begin with **WorkflowSystem**.

Some of the values that you see in drop-down lists in the user interface are not the same as the values that RICOH ProcessDirector uses internally. When you make some requests using Web services or set values using an overrides file, you must use the internal value. The **Internal values** column lists the internal values for those properties.

In the **Editable** column:

- Yes means that you can change the value.

- No means that you cannot change the value.

System properties

Database name	Field name	Brief description	Internal values	Editable
WorkflowSystem. AdLdap.TestUserName	LDAP test user name	Specifies the LDAP user name used to test the LDAP settings.		Yes
WorkflowSystem. AltSMTPAddress	Alternate SMTP server	Specifies either the network IP address or the fully-qualified host name of the SMTP server that RICOH ProcessDirector uses to send emails when the SMTP server type property on a job step is set to Alternate .		Yes
WorkflowSystem. AltSMTPPassword	Alternate SMTP password	Specifies the password that the primary server uses to log in to the alternate SMTP Server.		Yes
WorkflowSystem. AltSMTPPort	Alternate SMTP port	Specifies TCP/IP port that the alternate SMTP server uses.		Yes
WorkflowSystem. AltSMTPSSLPort	Alternate SMTP SSL port	If the alternate SMTP server uses SSL security, specifies the TCP/IP port that the SSL connection uses.		Yes
WorkflowSystem. AltSMTPTLSPort	Alternate SMTP TLS port	If the alternate mail server uses TLS security, specifies the TCP/IP port that the TLS connection uses.		Yes
WorkflowSystem. AltSMTPUserName	Alternate SMTP user name	Specifies the user name that the RICOH ProcessDirector		Yes

Database name	Field name	Brief description	Internal values	Editable
		server uses to log in to the alternate SMTP Server.		
WorkflowSystem.CaptureFileName	Capture file	Specifies the name of the capture file that is created when you capture system data.		Yes
WorkflowSystem.CaptureLevel	Data to capture	Shows the amount of information that is included in the capture file.		Yes
WorkflowSystem.CaptureServer	Servers to capture data from	Shows the server or servers that information is collected from when the capture file is created.		Yes
WorkflowSystem.Child.MaxJobID	Largest child job number	Specifies the largest job number that RICOH ProcessDirector can assign to a child job.		Yes
WorkflowSystem.Child.MinJobID	Smallest child job number	Specifies the smallest job number that RICOH ProcessDirector can assign to a child job.		Yes
WorkflowSystem.CopyTracesToTemp	Back up files before capture	Shows whether trace files are copied to a temporary directory before they are added to the capture file.		Yes
WorkflowSystem.CredentialExpiration	Login inactivity timer (minutes)	Specifies the number of minutes before an inactive user is automatically logged out.		Yes
WorkflowSystem.Email.SSLPort	Email SSL port	If the mail server uses SSL security, the TCP/IP port that the SSL connection uses.		Yes

Database name	Field name	Brief description	Internal values	Editable
WorkflowSystem.Email.TLSPort	Email TLS Port	If the mail server uses TLS security, the TCP/IP port that the TLS connection uses.		Yes
WorkflowSystem.FileSystemMapping	File system mapping file	Specifies the name of a file that maps file paths to mount points on the RICOH ProcessDirector server.		Yes
WorkflowSystem.GUILoggingLevel	Web server logging level	When the Custom option is selected for the Trace level property, shows the trace level that is active for the system.		Yes
WorkflowSystem.JobDisplayTag	Job identifier to use	Specifies whether the Printers portlet, printer console, and PSF Job completion log show the job name or the job number.	<ul style="list-style-type: none"> • Job.ID • Job.Name 	Yes
WorkflowSystem.LastCaptureFilename	File name for last capture	Specifies the name of the last capture file created when system data was most recently captured.		No
WorkflowSystem.LastCaptureTimestamp	Last capture completed	The date and time that the capture was most recently completed. RICOH ProcessDirector updates the value whenever the capture completes.		No
WorkflowSystem.LPDHostnames	Hosts allowed to submit LPD jobs	Specifies the host names or IP addresses of systems that can submit jobs to RICOH ProcessDirector using the LPD protocol.		Yes

Database name	Field name	Brief description	Internal values	Editable
WorkflowSystem. MaxJobID	Largest job number	Specifies the largest job number that RICOH ProcessDirector can assign to a job.		Yes
WorkflowSystem. MediaTolerance. RecyclingPercentage	Recycled content tolerance	Specifies the acceptable variation in recycled content for a media object.		Yes
WorkflowSystem. MediaTolerance.Size	Size tolerance	Specifies the acceptable variation in size for a media object.		Yes
WorkflowSystem. MediaTolerance.Weight	Weight tolerance	Specifies the acceptable variation in weight for a media object.		
WorkflowSystem. PrinterProgress	Print progress bar	Specifies whether the print progress bar is displayed in the Printers portlet.	<ul style="list-style-type: none"> • No • Yes 	Yes
WorkflowSystem. MaxPasswordAge	Maximum password age before expiration	Specifies the number of days passwords can be used until they expire.		Yes
WorkflowSystem. MinJobID	Smallest job number	Specifies the smallest job number that RICOH ProcessDirector can assign to a job.		Yes
WorkflowSystem. Primary. MaxLowUsageSteps	Maximum step count for other steps	Controls how many non-resource-intensive steps the primary RICOH ProcessDirector server lets run concurrently.		Yes
WorkflowSystem. Primary. MaxHighUsageSteps	Maximum resource-intensive step count	Controls how many resource-intensive steps the primary RICOH ProcessDirector server lets run concurrently.		Yes

Database name	Field name	Brief description	Internal values	Editable
WorkflowSystem. ProxyPassword	Proxy server 1 password	Specifies the password that the primary server uses to log in to proxy server 1.		Yes
WorkflowSystem. ProxyPassword2	Proxy server 2 password	Specifies the password that the primary server uses to log in to proxy server 2.		Yes
WorkflowSystem. ProxyPort	Proxy server 1 port	Specifies TCP/IP port that RICOH ProcessDirector uses to communicate with proxy server 1.		Yes
WorkflowSystem. ProxyPort2	Proxy server 2 port	Specifies TCP/IP port that RICOH ProcessDirector uses to communicate with proxy server 2.		Yes
WorkflowSystem. ProxyServer	Proxy server 1	Specifies either the network IP address or the fully qualified host name of the proxy server 1. RICOH ProcessDirector uses a proxy server to connect to external web sites.		Yes
WorkflowSystem. ProxyServer2	Proxy server 2	Specifies either the network IP address or the fully qualified host name of the proxy server 2. RICOH ProcessDirector uses a proxy server to connect to external web sites.		Yes
WorkflowSystem. ProxyUser	Proxy server 1 user	Specifies the user name that RICOH ProcessDirector uses when proxy server 1 connects to an external web site.		Yes

Database name	Field name	Brief description	Internal values	Editable
WorkflowSystem. ProxyUser2	Proxy server 2 user	Specifies the user name that RICOH ProcessDirector uses when proxy server 2 connects to an external web site.		Yes
WorkflowSystem. RememberPrinterStatus	Remember enabled status of printers	Specifies whether RICOH ProcessDirector restores the state of enabled printers after a system shutdown.	<ul style="list-style-type: none"> • No • Yes 	Yes
WorkflowSystem. RetainPollInterval	Retention polling interval (minutes)	Controls how often RICOH ProcessDirector polls jobs that are retained on the system to determine if any further action is necessary.		Yes
WorkflowSystem. SMTPPassword	SMTP password	Specifies the password that RICOH ProcessDirector uses to login to the SMTP Server.		Yes
WorkflowSystem. SMTPPort	SMTP port	Specifies the TCP/IP port that the SMTP server uses.		Yes
WorkflowSystem. SMTPServer	SMTP server	Specifies either the network IP address or the fully-qualified host name of the SMTP server that RICOH ProcessDirector uses to send email.		Yes
WorkflowSystem. SMTPUsername	SMTP user name	Specifies the user name that RICOH ProcessDirector uses to connect to the SMTP Server.		Yes
WorkflowSystem. SystemIdentifier	System identifier	Specifies an alternate name for		Yes

Database name	Field name	Brief description	Internal values	Editable
		the system to show in the user interface.		
WorkflowSystem. TraceCustomTraceLevel	Custom trace level	When the Custom option is selected for the Trace level property, shows the trace level that is active for the system.		Yes
WorkflowSystem. TraceFileCount	Maximum number of trace files	Shows the maximum number of trace files that can be created on the system.		Yes
WorkflowSystem. TraceFileSize	Maximum trace file size	Shows the upper size limit in kilobytes (KB) for trace files that the system creates.		Yes
WorkflowSystem. TraceLevel	Trace level	Shows the level of tracing that is active on the system.		Yes

Database property names for users

Messages about users and their passwords might refer to properties by their database names, which begin with **User**.

In the **Editable** column:

- Yes means that a user can change the value after the user is created.
- No means that a user cannot change the value.

User properties

Database name	Field name	Brief description	Editable
User.Description	User description	Contains text that describes the user name.	Yes
User.Groups	Group membership	Contains the name of the group or groups that the user belongs to.	Yes
User.ID	Displayed in the page title	Contains the user name for an authorized user of RICOH ProcessDirector.	No
User.LastModified	Last Modified	The date and time that the user was last changed.	No

Database name	Field name	Brief description	Editable
		RICOH ProcessDirector updates the value whenever the user changes.	
User.Login.Status	Logged in	Specifies whether a user is logged in to RICOH ProcessDirector.	No
User.LocationFilterPreference	Locations to show	Shows which locations are currently used to filter the user interface. Only objects from locations selected in this list are displayed.	Yes
User.LocationSettings	Allowed locations	Several objects such as jobs, printers, and input devices have a Location property. The Allowed locations property lets you specify which locations a user can access. The setting affects which objects the user can see in the user interface. The user can select which location to use to filter the user interface using the Locations to show property.	Yes
User.ModifiedBy	Modified by user	Specifies the user who made the last change to this input device. RICOH ProcessDirector maintains the value of this property, and updates the value whenever the input device changes.	No
User.Password	User password	Accepts the password for the user.	Yes
User.PasswordConfirm	Confirm new user password	Specifies the password for the new user. Must match the New user password value.	Yes
User.PasswordLastChanged	Password last changed	Specifies the date and time when the password was last changed.	No

Database name	Field name	Brief description	Editable
User.PasswrodNew	New user password	Specifies the password for the new user.	Yes
User.StartPage	Start page	Specifies which page RICOH ProcessDirector displays after you log in. When a new user is created, this control is not available until a Group membership value is selected.	Yes

Database property names for groups

Messages about security groups might refer to properties by their database names, which begin with **Group**.

In the **Editable** column:

- Yes means that a user can change the value after the group is created.
- No means that a user cannot change the value.

Group properties

Database name	Field name	Brief description	Editable
Group.Actions	Allowed actions	Lists the actions that users in this group are authorized to do.	Yes
Group.Attributes	Allowed attributes	Lists the object properties that users in this group are authorized to edit.	Yes
Group.Description	Group description	Contains text that describes the group.	Yes
Group.ID	Displayed in the page title	Contains the name of the group.	No
Group.SourceID	Group source ID	Contains the name of the group that this group was copied from.	No

Database property names for locations

Messages about locations might refer to properties by their database names, which begin with **Location**.

In the **Editable** column:

- Yes means that an authorized user can change the value after the location has been created.

- No means that an authorized user cannot change the value.

Location properties

Database name	Field name	Brief description	Editable
Location.Description	Location description	Describes the location. For example, the description might include the city name or the building name and floor where printers are located.	Yes
Location.ID	Location name	Specifies the name of the location.	No
Location.LastModified	Last modified	The date and time that the location was last changed.	No
Location.Modified	Modified by user	Specifies the user name of the user who made the last change to this location.	No

Database property names for security

Messages about security might refer to properties by their database names, most of which begin with **WorkflowSystem**. The database property names for security include the database property names for Lightweight Directory Access Protocol (LDAP).

In the **Editable** column:

- Yes means that an authorized user can change the value after the initial security setup.
- No means that an authorized user cannot change the value.

Security properties

Database name	Notebook tab: Field name	Brief description	Internal values	Editable
User.AccountStatus	Security → User-s : Account status	Shows the current status of the user account.	<ul style="list-style-type: none"> • Active • Locked-Inactive • Locked-Password Failure 	Yes
WorkflowSystem. AdLdap.EmailAddress	Security → Settings → LDAP : Email attribute	Specifies the LDAP attribute that RICOH ProcessDirector gets user email addresses from.		Yes
WorkflowSystem. AdLdap.GroupMap	Security → Settings → LDAP : Product to LDAP group mapping	Specifies the mapping of RICOH ProcessDirector groups to LDAP groups.		Yes

Database name	Notebook tab: Field name	Brief description	Internal values	Editable
WorkflowSystem. AdLdap. GroupSearchBase	Security → Settings → LDAP: Group search base	Specifies the branch of the LDAP external directory tree that RICOH ProcessDirector searches to identify the organizational unit (OU) that contains LDAP groups.		Yes
WorkflowSystem. AdLdap. GroupSearchFilter	Security → Settings → LDAP: Group search filter	Specifies the filter that RICOH ProcessDirector uses to determine if a group with the specified name exists within the LDAP group search base.		Yes
WorkflowSystem. AdLdap. GroupSearchMember	Security → Settings → LDAP: Group search member	Specifies the LDAP attribute RICOH ProcessDirector uses to get the distinguished names of LDAP group members.		Yes
WorkflowSystem. AdLdap.ManagerDN	Security → Settings → LDAP: Manager distinguished name	Specifies the full distinguished name (DN) of the user that binds to the LDAP server for user searches.		Yes
WorkflowSystem. AdLdap. ManagerPassword	Security → Settings → LDAP: Manager distinguished name password	Specifies the password for the user listed in the Manager distinguished name property.		Yes
WorkflowSystem. AdLdap.rootDN	Security → Settings → LDAP: Root distinguished name	Specifies the distinguished name (DN) that is not limited by administrative limit restrictions or access control restrictions for the database.		Yes

Database name	Notebook tab: Field name	Brief description	Internal values	Editable
WorkflowSystem. AdLdap.Server	Security → Settings → LDAP : LDAP server	Specifies either the network IP address or the fully-qualified host name of one or more LDAP servers and the ports that the system uses for authentication.		Yes
WorkflowSystem. AdLdap.SyncLdapGrp	Security → Settings → LDAP : Synchronize with LDAP groups	Specifies if RICOH ProcessDirector updates the product security groups for a user, based on the values for the Product to LDAP group mapping property, each time the user logs in.	<ul style="list-style-type: none"> • Yes • No 	Yes
WorkflowSystem. AdLdap.UserSearchBase	Security → Settings → LDAP : User search base	Tells the server which part of the external directory tree to search, relative to the base distinguished name (DN) in the LDAP URL.		Yes
WorkflowSystem. AdLdap.UserSearchFilter	Security → Settings → LDAP : User search filter	Specifies the filter that RICOH ProcessDirector uses to determine if a distinguished name (DN) of the user exists in LDAP.		Yes
WorkflowSystem. AdLdap.YesNo	Security → Settings → LDAP : Authenticate with LDAP	Specifies whether users can log in to RICOH ProcessDirector with a user ID that is defined in the existing Lightweight Directory Access Protocol (LDAP) server.	<ul style="list-style-type: none"> • Yes • No 	Yes
WorkflowSystem. ComplexRules	Security → Settings → General : Enforce	Specifies whether all users must use complex passwords.	<ul style="list-style-type: none"> • Yes • No 	Yes

Database name	Notebook tab: Field name	Brief description	Internal values	Edita- ble
	password complexity rules			
WorkflowSystem. InactiveLength	Security → Setti- ngs → General : Account inactivity period	Specifies the number of days that RICOH ProcessDirector lets a user account be inactive before that user is locked out of the system.		Yes
WorkflowSystem. LockOutLength	Security → Setti- ngs → General : Lockout duration	Specifies the amount of time that RICOH ProcessDirector locks a user out of the system after the user exceeds the Account login threshold. The unit of time for the value can be minutes, hours, or days. Use the toggle control to the right of the property name to select the unit of time for the value.		Yes
WorkflowSystem. MaxLoginAttempts	Security → Setti- ngs → General : Account lockout threshold	Specifies the number of unsuccessful login or password change attempts that are allowed before the user is locked out.		Yes
WorkflowSystem. MinPasswordLength	Security → Setti- ngs → General : Minimum password length	Specifies the minimum number of characters required for a password.		Yes
WorkflowSystem. PasswordReuseCount	Security → Setti- ngs → General : Password reuse count	Specifies how many times a user must enter a unique password before they can reuse an old password.		Yes

9. Job properties that can be set from the job ticket

When you submit a job with a job ticket, RICOH ProcessDirector sets some job properties from values in the job ticket.

Job properties are mapped to attributes in the job ticket. In most cases, RICOH ProcessDirector sets the job property to a value that corresponds to the value of the attribute in the **Job ticket attribute** column. (The values are not always identical.) The **Notes** column explains variations in this process.

When a job ticket refers to more than one print file, RICOH ProcessDirector creates a child job for each file. In the **Per job or per ticket** column:

- **Job** means that the property is mapped to an attribute in the job ticket that applies to individual print files, so it can have a different value for each child job.
- **Ticket** means that the property is mapped to an attribute that applies to the entire job ticket, so it must have the same value for all jobs created for the job ticket.
- **Job or ticket** means that the property is mapped to both types of attribute, so it can have the same value or different values depending on the attributes in the job ticket.

Job properties that can be set from the job ticket

Database name	Note-book tab: field name	Brief description	Per job or per ticket	JDF attribute name	Notes
Job.Binding	Binding	Set the binding settings for the job you are printing.	Job or ticket	job-binding	
Job.Copies	General and Status: Job copies requested	Contains the number of job copies that have been requested.	Job or ticket	job-copies	
Job. Customer-Name	Scheduling: Customer name	Identifies the customer who is associated with this job.	Job or ticket	job-contact-info	
Job. Description	General: Job description	Contains text that describes the job.	Job or ticket	jt-comment	
Job.Duplex	General: Duplex	Indicates whether duplexed printing is active for the job and, if	Job or ticket	job-sides	

Database name	Note-book tab: field name	Brief description	Per job or per ticket	JDF attribute name	Notes
		so, the type of duplexed printing.			
Job.FoldOptions	Fold Options	Specifies how to fold the job or large sheets in the job, whether to fold all the sheets together (overlap or collate), and whether the front side of the sheet ends up on the outside rather than the inside of the fold.	Job or ticket	job-folding	
Job.Info.Department	Information: Department information	Contains a department description for the job.	Job or ticket	job-contact-info	
Job.InputData-stream	General: Input data stream	Specifies the format of data that the input file for this job contains.	Job	document-format	
Job.Line-2AFP.CC_TYPE	AFP: Carriage control type	Indicates the type of carriage controls that are present in the job.	Ticket	job-carriage-control-characters	This property is only available if the AFP Support Feature is installed.
Job.Line-2AFP.FORMDEF	AFP and Print: Form definition	Identifies the form definition to	Ticket	job-form-definition	This property is only available if the AFP Support Feature is installed.

Database name	Note-book tab: field name	Brief description	Per job or per ticket	JDF attribute name	Notes
		use with the job.			
Job.Line-2AFP.PAGEDEF	AFP: Page definition	Identifies the AFP page definition to use with the job.	Ticket	job-page-definition	This property is only available if the AFP Support Feature is installed.
Job.Line-2AFP.TRC	AFP: Table reference characters	Indicates whether table reference characters are present in the job.	Ticket	job-table-reference-characters	This property is only available if the AFP Support Feature is installed.
Job.Media	Scheduling: Media (ready supported all)	Specifies the media to use for the job.	Job or ticket	The job media database name maps to several job-media and document-media attributes in JDF.	RICOH ProcessDirector uses the media detection setting to determine how this value is set. If a job ticket specifies both page-level and job-level media values, multiple values are set for this property. However, if you update the value of the Media property, it cannot be written back to the job ticket, because you cannot indicate the page range that the media should be used for. The media values in the job ticket remain unchanged.
Job.Name	General: Job name	Contains the name of the job.	Job or ticket	job-name	
Job.OutputBin	Scheduling: Output bin (requested	Specifies the output bin to use for the job.	Job or ticket	document-output-bin-name, job-output-bin-name	

Database name	Note-book tab: field name	Brief description	Per job or per ticket	JDF attribute name	Notes
	available all)				
Job.Print. JogCopies	Print: Jog output copies	Controls whether the printer jogs the output copies for the job.	Ticket	job-jog-offset	
Job.Print. Xoffset	Print: X offset (<i>unit</i>)	Identifies the offset, in inches or millimeters, in the x or horizontal direction of the logical page origin from the media origin.	Job or ticket	image-shift-front-x	This value applies to both the front and back of the page. This property is only available if the AFP Support Feature is installed.
Job.Print. Yoffset	Print: Y offset (<i>unit</i>)	Identifies the offset, in inches or millimeters, in the y or vertical direction of the logical page origin from the media origin.	Job or ticket	image-shift-front-y	This value applies to both the front and back of the page. This property is only available if the AFP Support Feature is installed.
Job.Priority	Schedul- ing: Job priority	Contains the printing priority of the job.	Job or ticket	job-priority	
Job.Punch	Schedul- ing: Punch	Specifies the number and position of holes to punch in the output.	Job or ticket	document-hole- making, job-hole- making	

Database name	Note-book tab: field name	Brief description	Per job or per ticket	JDF attribute name	Notes
Job.Requested-Printer	Scheduling: Requested printer	Contains the name of the printer that was requested for the job.	Job or ticket	job-logical-destination-name	
Job.RetainDuration	General: Retention period (<i>unit</i>)	Controls the amount of time in minutes, hours, or days that RICOH ProcessDirector retains a job after it reaches the RetainCompletedJobs step.	Job or ticket	job-retain	
Job.Staple	Scheduling: Staple	Specifies the number and position of staples to use in the output.	Job or ticket	job-stitching, document-stitching	

