

Configuring

[Configuring](#) / [Preparing for job submission](#) / [Preparing to submit batch jobs](#) / [Batching methods](#) / Pattern

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
<code>.*\$</code>	<code>.*jdf\$</code>	overrides	Other	jdf	abc000317.jdf	/aiw/aiw1/spool/default/10000143/1000143overrides.jdf	JDF job ticket
<code>.*\$</code>	<code>.*oth\$</code>	overrides	Other	txt	abc00317.oth	/aiw/aiw1/spool/default/10000143/1000143overrides.txt	Overrides file
<code>.*\$</code>	<code>.*pdf\$</code>	print	Data	pdf	abc00317.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	<code>(.*)\..pdf</code>	The expression <code>(.)</code> defines a backreference to the data file name without the extension. When a data file name matches this pattern, all the characters that match <code>(.)</code> are assigned to backreference <code>\1</code> .
Batching	File pattern	<code>\1\.jdf</code>	Matches the data file name, but changes the extension to <code>.jdf</code> . <code>Job1.pdf</code> and <code>Job1.jdf</code> are considered a match and are included in the job, but <code>Job2.jdf</code> does not match.

In a more complex example:

Tab	Field	Value	Description
Batching	Data patterns	(abc) (def) \.pdf, .*-(12)-.*\.\pdf	The data file must match one of these two comma-separated patterns. Data pattern 1: \1 = abc and \2 = def The expressions (abc) and (def) define backreferences to the data file name. The extension must be .pdf. A data file that matches pattern 1 is abcdef.pdf. Data pattern 2: \1 = 12 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 year-month-day). The extension must be .pdf. A data file that matches pattern 2 is 2011-12-02.pdf.
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.

Parent topic: [Batching methods](#)