Configuring



# Setting up a workflow that processes a pull list

To extract documents from a job using a pull list, you set up a workflow that includes a step based on the **SetDocPropsFromList** step template. Additional steps and conditional processing (or additional workflows) are required to process both the documents extracted from the job and the documents remaining in the job.

The **PullPDFSample** supplied workflow resembles the **PDF** workflow that we are going to set up in this procedure. Before you set up your own workflow, open the **PullPDFSample** workflow. Examine the steps and conditional processing in the workflow as you read this procedure.

**Note:**
- If you have the AFP Support feature installed, the **PullAFPSample** supplied workflow shows how to extract documents from a workflow that processes AFP files.

To set up a workflow that processes a pull list:

1. Identify data that determines whether a document is pulled.
   Examples:
   - You want to pull documents based on a list of account numbers. The account number determines whether a document is pulled.
   - You want to pull documents based on a list of postal codes. The postal code determines whether a document is pulled.
   - You want to pull documents based on a list of policy types and states. The policy type and state determine whether a document is pulled.
2. Decide which document properties you are going to use to specify the data that the **IdentifyPDFDocuments** step (PDF files) or **IdentifyDocuments** step (AFP files) extracts from each document in the print file.
   You can use existing RICOH ProcessDirector document properties, or you can define your own custom document properties.

   Examples:
   - You can use the **Doc.PullProp** document property, which is supplied with all document processing features.
   - You can use one or more custom document properties, such as **Doc.Custom.AccountNumber**, **Doc.Custom.PostalCode**, **Doc.Custom.PolicyType**, and **Doc.Custom.State**.
     If the custom document properties do not already exist, you must define them.

     **Note:**
     - To define document properties:
       - Edit the document properties configuration file.
       - Run the **docCustom** utility.
         The first time that you run the utility, it creates the Custom Document Properties feature.
       - Use Feature Manager to install or update the Custom Document Properties feature.
       - If you are working with PDF files, load the updated RICOH ProcessDirector document properties to RICOH ProcessDirector Plug-in for Adobe Acrobat.
3. Specify the data that the **IdentifyPDFDocuments** step (PDF files) or **IdentifyDocuments** step (AFP files) extracts from each document in the job:
   - If you are working with PDF files, use the **Define Document Property** function in RICOH ProcessDirector Plug-in for Adobe Acrobat.
   - If you are working with AFP files, use the Document Property Designer (DPD) mode of RICOH Visual Workbench.

     **Note:**
     - If the AFP files do not have index tags defined for the document data that you want to extract, use AFP Indexer to add the tags. AFP Indexer is installed with the AFP Support feature.
4. Save your document property definitions in a control file for use with the **IdentifyPDFDocuments** step (PDF files) or **IdentifyDocuments** step (AFP files):
   - If you are working with PDF files, use the **Save control file** function in RICOH ProcessDirector Plug-in for Adobe Acrobat.
   - If you are working with AFP files, use the **Save control file** function in RICOH Visual Workbench.
5. Send the control file to the RICOH ProcessDirector server in a directory that the RICOH ProcessDirector system user has access to.
6. Log in to RICOH ProcessDirector.
7. Click the **Workflow** tab.
8. Make a copy of the workflow that you want to modify, or create a new workflow.

   In this procedure, we modify a simple workflow that processes PDF files. The workflow has these steps:

- SetJobPropsFromTextFile
- CountPages
- IdentifyPDFDocuments
- WriteDocumentsToDatabase
- BuildPDFFromDocuments
- UpdateDocumentsInDatabase
- CreatePageRanges
- PrintJobs
- RetainCompletedJobs
- RemoveJobs

> **Note:**
> - If you have the AFP Support feature installed, a simple workflow that modifies AFP files might have **UseInlineFormDefinition** and **EnableRepositioning** steps in place of the **CountPages** step, an **IdentifyDocuments** step in place of the **IdentifyPDFDocuments** step, and a **BuildAFPFromDocuments** step in place of the **BuildPDFFromDocuments** step.

9. Add a step based on the **SetDocPropsFromList** step template to the workflow after the **WriteDocumentsToDatabase** step.
10. Set values for the properties of the **SetDocPropsFromList** step:
    1. For the **List file directory** property, specify the location of the directory that contains the pull lists.

       For example: `/aiw/aiw1/clientfiles/pull`.
    2. For the **Delimiter** property, specify the delimiter used to separate values in the pull list.
       If the pull list only uses one property, you must put each value on a separate line and specify **New Line** as the delimiter.
       If the pull list uses two or more properties, you must put each set of values on a separate line. Specify the delimiter that you use to separate the values on each line: **Tab**, **Semicolon**, **Comma**, **Space**, or **Tilde**.

       Examples:
       - The pull list contains account numbers:
         ```
         4377852A
         4372341A
         4400076A
         4401132H
         ```
         Set the value of the **Delimiter** property to **New Line**.
       - The pull list contains policy types and states separated with commas:
         ```
         Home,AZ
         Home,CO
         Auto,CO
         ```
         Set the value of the **Delimiter** property to **Comma**.
    3. For the **Columns in list file** property, select all the document properties that you are using to specify the data that determines whether a document is pulled.

       Examples:
       - **Doc.PullProp**
       - **Doc.Custom.AccountNumber**
       - **Doc.Custom.PostalCode**
       - **Doc.Custom.PolicyType** and **Doc.Custom.State**
    4. If the you are using two or more document properties to specify the data, order the document properties (from top to bottom) to match the order of the data columns in the pull list (from left to right).

       To rearrange the properties, click  , the pencil icon. In the dialog that opens, right-click each selected property and choose **Move to top**. After all of the selected properties are at the top of the list, click and drag them into the correct order. After rearranging the properties, click outside of the dialog to close it.
    5. For the **Stop for excess columns** property:
       - Select **YES** if you want the step to move into an error state if the pull list has more columns of data than the number of document properties specified by the **Columns in list file** property.
         For example, select **YES** if the pull list has two columns of data and the **Columns in list file** property specifies two document properties. You want the step to go into error if a pull list with four columns of data is placed in the list file directory.
       - Select **NO** if you do not want the step to move to an error state if the pull list has more columns of data than the number of document properties specified by the **Columns in list file** property.
         For example, select **NO** if the pull list has four columns of data but you are using only the first column. The **Columns in list file** property specifies one document property.

         > **Note:**
         > - If the pull list has excess columns, they must all be to the right of columns you are using.

       The step always moves into an error state if the pull list has fewer columns of data than the number of properties specified by the **Columns in list file** property.

6. For the **Document property to set** property, select the document property that you want to use to specify whether a document is pulled.
   The **Doc.Pull** property is supplied with all document processing features. It is a convenient choice for the value of the **Document property to set** property. As an alternative, you can create a custom document property or use an existing document property as the value of **Document property to set**.

   > ★ **Important:**
   > - If you use a document property that already contains a value for the documents in the job, RICOH ProcessDirector overwrites the original value with the new value for matching documents or the new value for other documents. Because the new value replaces the original value, make sure that you no longer need the original value.

7. Specify values for the **Value for matching documents** and **Value for other documents** properties.
   If the document property that you specify as the value of the **Document property to set** property does not exist in the document properties file for the job, RICOH ProcessDirector creates a column for the document property in the file and populates the column with values specified for the **Value for matching documents** and **Value for other documents** properties.
   If the document property does exist in the document properties file, RICOH ProcessDirector changes the values for the property based on the values of the **Value for matching documents** and **Value for other documents** properties.

   Example:
   - A document properties file contains three document properties:

     | Doc.Custom.AccountNumber | Doc.Custom.PolicyType | Doc.Custom.State |
     |---|---|---|
     | 144372 | Home | CO |
     | 144372 | Auto | CO |
     | 144372 | Business | CO |
     | 187456 | Home | AZ |
     | 187456 | Auto | AZ |
     | 187456 | Business | AZ |
     | 223114 | Home | NY |
     | 223114 | Auto | NY |
     | 223114 | Business | NY |

   - A pull list contains values for the **Doc.Custom.PolicyType** and **Doc.Custom.State** document properties:

     ```
     Home,AZ
     Home,CO
     Auto,CO
     ```

   - The value of the **Document property to set** property is **Doc.Pull**. The document properties file for the job does not have a column for the **Doc.Pull** document property.
   - The value of the **Value for matching documents** property is **YES**.
   - The value of the **Value for other documents** property is **NO**.
   - When a job enters the **SetDocPropsFromList** step, RICOH ProcessDirector:
     - Creates a column for **Doc.Pull** in the document properties file for the job.
     - Sets the value of the **Doc.Pull** document property to **YES** if the values of the document properties for a document match all the values of the document properties on the pull list.
       The value of the **Doc.Pull** document property is set to **YES** for documents containing Home policies in Arizona (AZ) and documents containing Home or Auto policies in Colorado (CO).
     - Sets the value to **NO** if the value of any document property for a document does not match the value of a document property on the pull list.
       The value of the **Doc.Pull** document property is set to **NO** for documents containing Auto policies in Arizona, documents containing any policy other than Home or Auto, and documents containing policies in any state except Arizona or Colorado.
     - The updated document properties file has four document properties:

       | Doc.Custom.AccountNumber | Doc.Custom.PolicyType | Doc.Custom.State | Doc.Pull |
       |---|---|---|---|
       | 144372 | Home | CO | YES |
       | 144372 | Auto | CO | YES |
       | 144372 | Business | CO | NO |
       | 187456 | Home | AZ | YES |
       | 187456 | Auto | AZ | NO |
       | 187456 | Business | AZ | NO |
       | 223114 | Home | NY | NO |
       | 223114 | Auto | NY | NO |
       | 223114 | Business | NY | NO |

8. Edit the other step properties as needed.

11. **Optional:** If you want jobs to wait until you receive a pull list, add a **Wait** step to the workflow before the **SetDocPropsFromList** step. Specify values for the step properties.

    Examples:
    - To wait until 6 PM, set the **Wait until** property to `6:00 PM`. Do not specify values for the **Wait for** and **Complete step after** properties.
    - To wait four hours, set the **Wait for** property to `4 hours`. Do not specify values for the **Wait until** and **Complete step after** properties.
    - To wait six hours or until 5 PM, whichever occurs first, set the **Wait until** property to `5:00 PM`, the **Wait for** property to `6 hours`, and the **Complete step after** property to **First occurs**.

- To wait at least three hours and at least until 4 PM, whichever occurs last, set the **Wait until** property to `4:00 PM`, the **Wait for** property to `3 hours`, and **Complete step after** property to **Last occurs**.

12. Add steps that process the documents after the **SetDocPropsFromList** step.

    For example:
    - You can add a **GroupDocuments** step and set the value of the **Group first** property to **Pull document**.
    - Then you can add a **CreateJobsFromDocuments** step and set the value of the **Child workflow** property to the name of the current workflow.

      > **Note:**
      > - This example uses the conditional processing in the **PullPDFSample** supplied workflow. As an alternative, you can set the **Child workflow** property to the name of another workflow and use that workflow to process the child jobs.

    - The **CreateJobsFromDocuments** step creates two child jobs: one for the group of documents with **Pull document** set to **YES** and another for the group of documents with **Pull document** set to **NO**.

13. Add conditional processing for documents that are pulled and documents that remain in the job.

    For example:
    - Add conditional processing for parent and child jobs near the start of the workflow, after the **SetJobPropsFromTextFile** step.
      - Define a rule for the branch that receives the parent jobs:
        **Job number Unlike *.***
        In our example, this branch is connected to the existing **DetectInputDataStream** step.
      - Add a step based on the **SetDocPropsFromConditions** step template.
      - Create a new branch for the child jobs, which have a decimal point in their job number, and connect the branch to the **SetDocPropsFromConditions** step.
        In our example, connect the **SetDocPropsFromConditions** step to the **BuildPDFFromDocuments** step, which connects to the **UpdateDocumentsInDatabase** step.
    - Set the properties for the **SetDocPropsFromConditions** step.
      In our example, the step assigns a value to a job property based on the value of the **Pull document** document property. The property conditions file sets the value of the **Custom 1** job property (database property name **Job.Info.Attr1**) to `Pull` or `Print`. This example shows the contents of the property conditions file:

      ```
      "Doc.Pull","Job.Info.Attr1"
      "=YES","Pull"
      "=NO","Print"
      ```

14. Add steps that process the pulled documents.

    For example, if you are processing PDF jobs, you might add an **EmailDocuments** step that emails the pulled documents to someone for verification that the documents were pulled.

15. Add steps that process the documents to be printed.

    For example, if you are processing PDF jobs, you might add **CreatePageRanges** and **PrintJobs** steps.

16. Add conditional processing to send the child jobs to separate pull and print branches of the workflow.
    - In our example, add a connector between the **UpdateDocumentsInDatabase** step and the new **EmailDocuments** step. Add this rule to the connector: **Custom 1 = Pull**
    - In our example, add a connector between the **UpdateDocumentsInDatabase** step and the **CreatePageRanges** step. Add this rule to the connector: **Custom 1 = Print**

    When jobs are sent through the workflow and the child jobs reach the **SetDocPropsFromConditions** step, RICOH ProcessDirector sets the value of the **Custom 1** job property:
    - For a child job with the **Doc.Pull** property set to **Yes**, the **Custom 1** job property is set to `Pull`. The child job goes through the connector with the rule **Custom 1 = Pull**.
    - For a child job with the **Doc.Pull** property set to **No**, the **Custom 1** job property is set to `Print`. The child job goes through the connector with the rule **Custom 1 = Print**.

17. If you created conditional processing for parent and child jobs, send the parent and child jobs together to the **RetainCompletedJobs** step:
    1. Add a step based on the **WaitForRelatedJobs** step template to the workflow before the **RetainCompletedJobs** step.
    2. Connect the branch for parent jobs and the two branches for child jobs to the **WaitForRelatedJobs** step.

       In our example, connect these steps to the **WaitForRelatedJobs** step:
       - **CreateJobsFromDocuments** in the branch for parent jobs.
       - **EmailDocuments** in the branch for child jobs with documents that have been pulled.
       - **PrintJobs** in the branch for child jobs with documents to be printed.

    The workflow now resembles the **PullPDFSample** workflow with these differences:
    - **PullPDFSample** has **DetectInputDataStream**, **FailWithMessage**, **SetDocPropsFromOriginal**, **CountPagesChild**, **AssignJobValuesPull**, and **AssignJobValuesPrint** steps.
    - This workflow has an **EmailDocuments** step in place of the **AssignJobValuesPull** and **ManualStepWithAutoRestart** steps in the **PullPDFSample** workflow.

18. Save the workflow.

19. Test the workflow:
    1. Create one or more input devices to point to the workflow.

2. Enable the workflow.
3. Enable the input devices.
4. Place a sample pull list in the list file directory.
5. Submit your job to the input device.

Parent topic: Setting up to pull documents from a job