

Configuring

[Configuring](#) / [Setting up to exchange data with web services](#) / Running the sample workflow for processing orders retrieved from REST web services

Running the sample workflow for processing orders retrieved from REST web services

The **RestfulWebServiceWF** workflow shows how to process orders retrieved from REST web services that simulate a website for ordering books. A REST web service input device, a **CallRESTService** step, and a REST web service notification exchange data using the web services. The workflow also gives examples of these steps: **ConvertJSONToXML**, **ApplyXSLTransform**, **CreateJobsFromXML**, and **DownloadFile**.

This workflow uses these sample objects and files:

- Workflow: **RestfulWebServiceWF**
 - Input device: **RestfulWebServiceSample**
The input device makes a GET call to retrieve JSON orders from a sample REST web service at <http://localhost:15080/restapi/1.0/sample/order>
 - Notification: **RestfulWebServiceSampleNotify**
The notification makes a POST call to a sample web service at <http://localhost:15080/restapi/1.0/sample/completeJobTicket>
 - JSON orders
One order is retrieved every 30 seconds from the sample web service.
Contents of sample order:

```
{"Order": {"orderId": "ORDnumber", "customername": "Ricoh"}}
```

Each order contains a randomly generated *number*, for example: 1238875463.
 - Print files: **Brochure.pdf** and **Cover.pdf**
These files are downloaded locally from <http://localhost:15080/restapi/>
 - XSLT style sheet files:
 - **orderToOverrides.xslt**
 - **jobTicketToOverrides.xslt**
- The files are in the `/aiw/aiwl/testfiles/restsample` directory.
- Notification PDF files
A notification PDF file is generated for each completed order. Each file is named ORDER-*ORDnumber*, for example: ORDER-ORD1238875463.
 - The **CallRESTService** step makes a GET call to retrieve JSON job tickets from a sample REST web service at <http://localhost:15080/restapi/1.0/sample/jobTicket>

Note:

- The sample web services are available locally. They return sample data in response to specific requests made by the sample input device, notification, and **CallRESTService** step. The web services do not support other requests.
- The sample objects and files, including the **RestfulWebServiceWF** workflow, are installed with the Web Services Enablement feature.

To run the sample workflow:

1. Click the **Main** tab.
2. In the Printers portlet, right-click the **Sample** printer and select **Enable**.
3. In the Input Devices portlet, right-click the **RestfulWebServiceSample** REST web service input device and select **Enable and Connect**.
A job appears in the jobs table.
4. Right-click the **RestfulWebServiceSample** input device and select **Disable**.
5. Right-click the input device again and select **Disconnect**.

Note: The sample input device polls for orders every 30 seconds and retrieves a JSON job. If you do not disable and disconnect the input device, a new job appears in the jobs table every 30 seconds.

Each time the **RestfulWebServiceSample** input device retrieves a JSON job:

- The job is sent through the **Parent** branch of the workflow.
- The **SetJobPropsFromTextFile** step sets the **Customer name** property to **RicohSample** and the **Custom 1** property to **RicohCustom**.
- The **ConvertJSONToXML** step converts the job into XML.
- The **ApplyXSLTransform** step uses the `orderToOverrides.xslt` XSLT style sheet to convert 2 XML elements into 2 RICOH ProcessDirector job properties in an overrides file:

XML element	Job property
<code>orderId</code>	<code>Job.Info.Attr3</code>
<code>customername</code>	<code>Job.CustomerName</code>

Contents of sample overrides file:

`Job.Info.Attr3=ORD1238875463Job.CustomerName=Ricoh`

The step puts the overrides file in the spool directory for the job. RICOH ProcessDirector uses the values in the overrides file to set the values of the properties for the job.

- The **AssignJobValues** step sets the value of the **Job name** property to **ORDER-\$ {Job.Info.Attr3}**, for example: **ORDER-ORD1238875463**.
- The **CallRESTService** step makes a GET call to one of the sample web services that simulate the website for ordering books. The step specifies order ID as the value of the **Request parameters** property: `orderId:$ {Job.Info.Attr3}`.

The web service returns JSON job ticket information for the order to RICOH ProcessDirector.

Contents of sample job ticket information:

```
{"jobTicket": [{"itemnumber": "1182563839", "copies": "4", "media": "Letter Plain", "file": "http://localhost:15080/restapi/Brochure.pdf", "type": "Brochure", "title": "RPDBestSeller"}, {"itemnumber": "1934194376", "copies": "4", "media": "Letter Preprinted", "file": "http://localhost:15080/restapi/Cover.pdf", "type": "Cover", "title": "RPDBestSeller"}]}
```

- The **ConvertJSONToXML** step converts the JSON job ticket information into XML.

- The **CreateJobsFromXML** step uses the `//JobTicket` XPath expression to parse the XML job ticket information. The step finds 2 job tickets and creates 2 child jobs.

Contents of sample child jobs:

```
<?xml version="1.0" encoding="UTF-8"?><JobTicket><copies>4</copies><file>http://localhost:15080/restapi/Brochure.pdf</file>
<itemnumber>2112076335</itemnumber><media>Letter Plain</media><type>Brochure</type><title>RPDBestSeller</title></JobTicket><?xml version="1.0"
encoding="UTF-8"?><JobTicket><copies>4</copies><file>http://localhost:15080/restapi/Cover.pdf</file><itemnumber>1418780325</itemnumber><media>Letter
Preprinted</media><type>Cover</type><title>RPDBestSeller</title></JobTicket>
```

The step submits the 2 child jobs to the **RestfulWebServiceWF** workflow.

The rule on the **[2] Child** connector determines whether the jobs are child jobs by checking for a decimal point in the job number. The connector sends child jobs through the **Child** branch of the workflow.

The original JSON job continues through the **Parent** branch to the **RunHotFolderApplication** step. That step puts the JSON job in a hot folder: `/aiw/aiwl/testfiles/restsample/sending`

One of the sample web services that simulate the website for ordering books pulls the hot folder for jobs. The web service processes each JSON job when it receives a request from the **RestfulWebServiceSampleNotify** notification. For more information, see the list item for the **RetainCompletedJobs** step.

- The **ApplyXSLTransform2** step uses the `jobTicketToOverrides.xslt` XSLT style sheet to convert 6 XML elements for each child job into RICOH ProcessDirector job properties in an overrides file:

XML element	Job property
<code>file</code>	<code>Job.Info.Attr2</code>
<code>itemnumber</code>	<code>Job.Info.Attr1</code>
<code>type</code>	<code>Job.Info.Attr4</code>
<code>title</code>	<code>Job.Info.Attr5</code>
<code>copies</code>	<code>Job.Copies</code>
<code>media</code>	<code>Job.Media</code>

Contents of sample overrides file:

`Job.Info.Attr2=http://localhost:15080/restapi/Brochure.pdfJob.Info.Attr1=1182563839Job.Info.Attr4=BrochureJob.Info.Attr5=RPDBestSellerJob.Copies=4Job.Me`

- The **DownloadFile** step downloads the `Cover.pdf` and `Brochure.pdf` files from <http://localhost:15080/restapi/> by setting the value of the **URL for download file** property to `${Job.Info.Attr2}`.

- The **AssignJobValues2** step sets the values of 3 job properties:

- Requested printer: Sample
- Input data stream: PDF
- Job name: \${Job.Info Attr5}-\${Job.Info Attr4}

The names of the child jobs for the `Brochure.pdf` and `Cover.pdf` files are now `RPDBestSeller-Brochure` and `RPDBestSeller-Cover`.

- The **OptimizePDF**, **CountPages**, **CreatePageRanges**, and **PrintJobs** steps process and print the `RPDBestSeller-Brochure` and `RPDBestSeller-Cover` child jobs on the **Sample** printer.

- When each child job arrives at the **RetainCompletedJobs** step, the **RestfulWebServiceSampleNotify** web service notification makes a `POST` call to one of the sample web services at `http://localhost:15080/restapi/1.0/sample/completeJobTicket`. The value of the **Request parameters** property is:

`jobTicket:${Job.Info Attr1}jobId:${Job.ID}`

The web service waits until all the child jobs for the original JSON job are in the **Retained** job state. The web service then creates a PDF file containing the job number and item number of each job ticket in the order. For example:

`10000001.2 with item number 278955095 has been processed``10000001.1 with item number 913895452 has been processed`

The web service puts the PDF file in the retrieval hot folder for the **RunHotFolderApplication** step: `/aiw/aiwl/testfiles/restsample/retrieval`

The **RunHotFolderApplication** step polls the hot folder, finds the PDF job, and sends the job to the next step.

- The **AssignJobValues** step sets the value of the **Input data stream** property to **PDF** and sends the PDF job to the **RetainCompletedJobs** step.

6. In the jobs table, right-click an **ORDER-ORDnumber** job and select **View**.

The viewer displays the PDF file. The information in the PDF file confirms that the 2 child jobs have been processed.

Parent topic: [Setting up to exchange data with web services](#)