Automation Engine

Integrating Automation Engine with Third Party Imposition Systems

ESKO✱
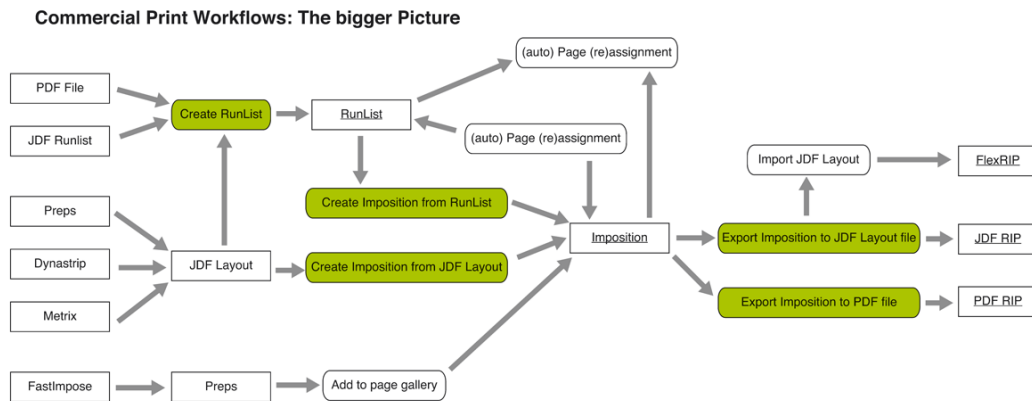
Contents

1. Introduction

There are new tasks in Automation Engine 12 to accommodate for third party impositioning programs. This gives flexibility in automation due to the source independence. These tasks import or export industry standard file types to or from the Impositions in the **Pages** and **Plates** views.

You can check or change (approval) statuses, assign pages and select elements for further processing or output in the **Pages** and **Plates** views in Automation Engine.

Note: You can find more info on this in the corresponding chapters of the User Guide.

2. Workflows: The bigger picture

You can find a schematic overview of how impositions can be created, processed and exported below. The tasks highlighted in green are new in Automation Engine 12.

Commercial Print Workflows: The bigger Picture

You can import populated or unpopulated impositions from applications that produces JDF layout file as its output. Some of the applications that can write a JDF Layout file are Kodak Preps, Litho Technics Metrix, Ultimate ImpoStrip , Dynagram Inpo2 , Dynagram Dynastrip, Plato and Pandora.

3. Importing and Creating Impositions

You can import the JDF Layout and use the **Create Imposition from JDF Layout** task to bring the **Imposition** to the **Pages** and **Plates** views in Automation Engine.

If you start with a PDF file or a JDF Runlist, you can use the **Create RunList** task to create a Runlist in Automation Engine. Subsequently, you can use the **Create Imposition from RunList** task to create Imposition from this RunList.

It is possible to separate the creation of a RunList and the creation of an Imposition from that Runlist in time. In some cases printers prefer to let the RunList be approved and corrected before they select the layout and create the Imposition.

You can achieve this by using two workflows.

1. Assemble the RunList and send it for approval.

2. Create the Imposition from the RunList and export it:

Read more about the new importing and creating tasks in:

Create Imposition from JDF Layout

Create RunList

Create Imposition from RunList
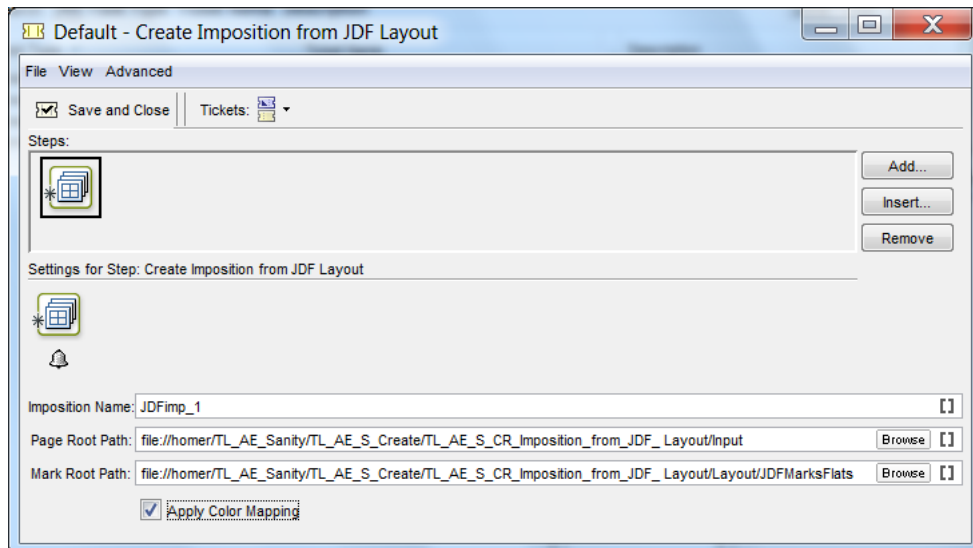
General Remarks

Note: File Compatibility

- JDF layout files: tests have been done with Preps, Metrix, ImpoStrip, InpO2, DynaStrip, Plato and Pandora.
- PDF files: Normalized as well as non Normalized files can be used in the Imposition workflows. However, third party rips and workflows do not support external images.

Note: Thumbnails in the **Pages** and **Plates** views are only shown when the pages in the files contain thumbnails. If you have no thumbnails in your PDF files, you can speed up the system.You can also save some screen space by hiding the details pane and by switching to **list** mode.
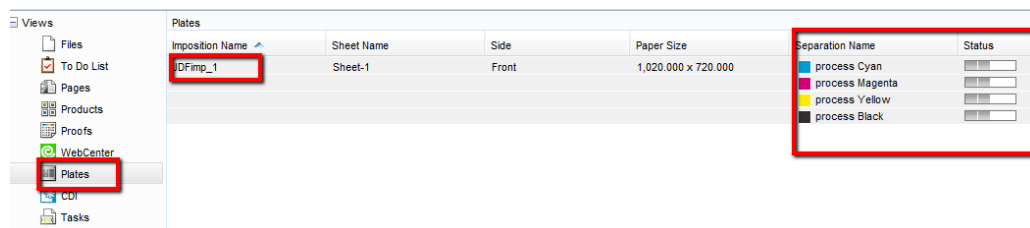
3.1 Create Imposition from JDF Layout

You can import a complete imposition including layout and page assignments into the **Pages** and **Plates** view using this task. This creates an IMP file from a JDF Layout file.

- **Imposition Name**: Here you can assign a name to the Imposition which will be visible in the Imposition Gallery.
- **Page Root Path**: You can specify the path to the pages referred in the JDF layout here.
- **Mark Root Path**: This path will be used as root to look for the marks reffered from the JDF Layout File.
- **Apply Color Mappings**: Enable this option when you want to use the same ink mapping scheme which is defined in the JDF file. This mapping can be viewed in **Plates** View.

Note:

- This option works only when the input JDF file contains a color mapping.
- The mapped inks will be visible in the **Plates** View after applying color mapping . However, the inks in the referenced input PDF files will remain the same as before.
- When you enable **Apply Color Mapping** in the **Export Imposition to PDF file** ticket, the color mapping from the **Plates** View will be exported to the output PDF file.
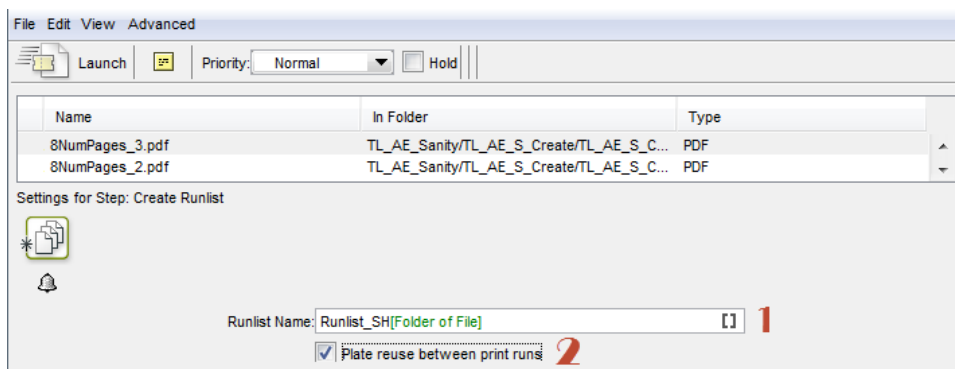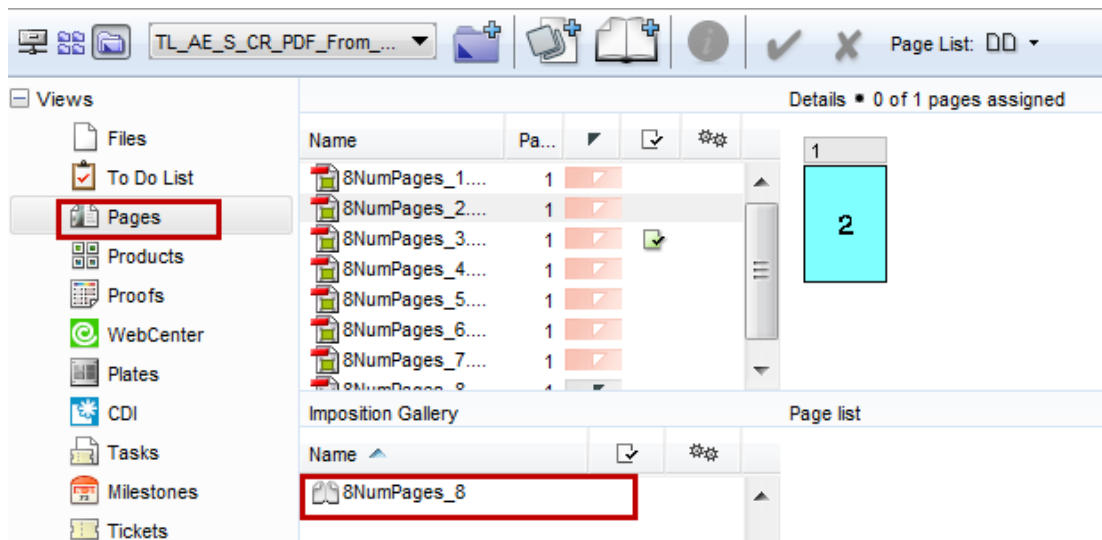
3.2 Create RunList

You can create RunList of a set of input files using this task. A **RunList** collects several PDF files sequentially to make one final JDF runlist (e.g. for imposition). It contains the names of the files to be collected or joined, their collecting / joining order and their original location. You can create a RunList from:

• imported JDF layout file
• imported JDF runlist
• one or more PDF files.

> **Note:** If you start from a JDF layout file, the layout part will be ignored.
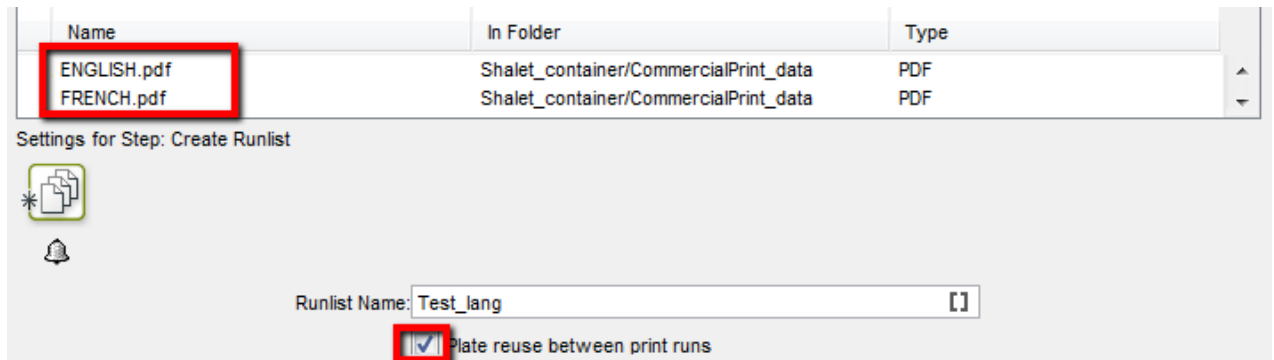

1. Give a suitable name in the **RunList Name** field. You can use SmartNames for this. You can see this name in the **Pages** view.
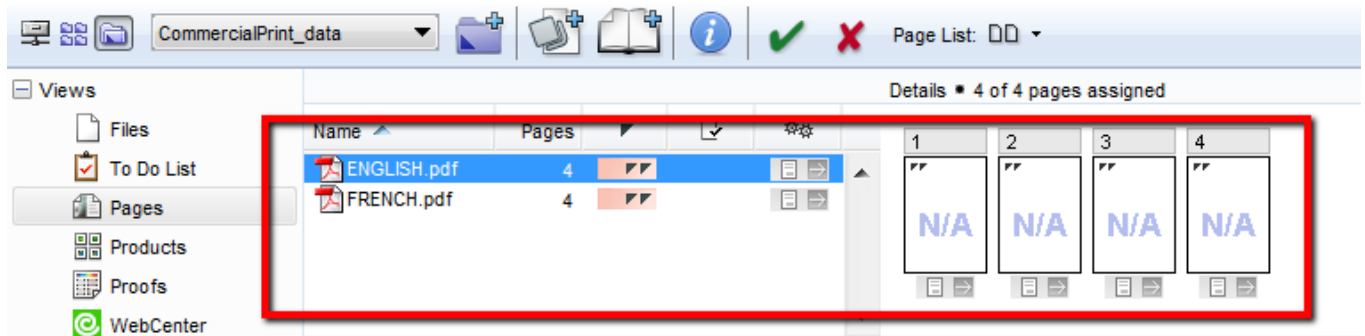

> **Note:** The tiles are outputted as multipage PDFs. When starting from a multipage PDF, one (multipage) PDF file will be generated for each page of the original file.

2. Enable **Plate reuse between runs** if you have language variants. This enables reuse of the image plates among the language variants. This option will consider each input file as a separate variant, if the number of pages in these files are equal.

You should use this in combination with the **Variant Separation** settings in **Job Setup**.

This will produce a RunList with multiple layers. Plates will be reused for common images in these language variants.

When you create an Impostion from a multilayer RunList, the system will take this into account when creating the plates in the **Plates** view.

3.3 Create Imposition from RunList

You can use this ticket in a workflow to create an Imposition from a RunList. This creates an IMP file from a RunList file.

- **Imposition Name**: You can use this name to identify the Imposition in the **Imposition Gallery**
- **Template**: You can browse to your JDF layout file here.

Note: You can use SmartNames like `nPages`, `pageHeight` and `pageWidth` to make the selection of the Layout Template dependent on the RunList.

- **Mark Root Path**: You have to enter the path to the file with marks referred from the JDF layout.
- **Fill remaining pages with blanks**:You can enable this if you want to add blank pages when the layout contains more pages than the RunList.

- **Remove RunList**: You can enable this if you want to remove the RunList after it is included in the Imposition.

You can view the results of this task in the **Imposition Gallery**.

4. Exporting Impositions

You can use **Export Imposition to JDF Layout** task to send an Impostition or part of an Imposition to another workflow system or to a Rip that is compatible with JDF layout.

- The output is JDF layout containing a JDF runlist.
- You can create a runlist for the **Join Pages** task using this output. This allows you to create a multipage PDF file to send to your customer for approval.
- It can be used to drive the FlexRip when followed by the **Import JDF Layout** task.

You can send an Impostition or part of an Imposition to a PDF Rip (e.g. Nexus Rip) using **Export Imposition to PDF** task. The output will be a PDF file containing the Imposition.

More info regarding these exporting tasks in:

Export Imposition to JDF Layout

Export Imposition to PDF File

4.1 Export Imposition to JDF Layout

Using this task, you can export Imposition, RunList, Layer, Plate and Page to JDF Layout and JDF Runlist format. This task will send an Imposition or a part of an Imposition to a workflow that requires JDF Layout as input. The JDF Layout contains a JDF RunList which you can use in the **Join Pages** task to produce a multipage PDF book for getting approval.

Apply Color Mapping: You can enable this to export the layout file with the same color mapping as defined in the **Plates** view.

Check page status: You can use this option to check if the approval status of the pages is compliant with the **Job setup** settings. The task will fail if it is not compliant.

4.2 Export Imposition to PDF File

This task exports Imposition, RunList, Layer, Plate and Page formats to PDF format.

Number of Flats per File: You can use this option to specify how you want to split the output file. One sheet side of an Imposition is considered as a Flat.

- If you enter `0` , the output will be one multipage file.
- If you enter `1` , the output will be two PDF files containing front(recto) and back(verso) sides separately. For each sheet in the Imposition, there will be two PDF files like this.
- If you enter `2` , the output will contain front(recto) and back(verso) side together in one PDF file with two pages. This will bethe outcome for each sheet in the Imposition .

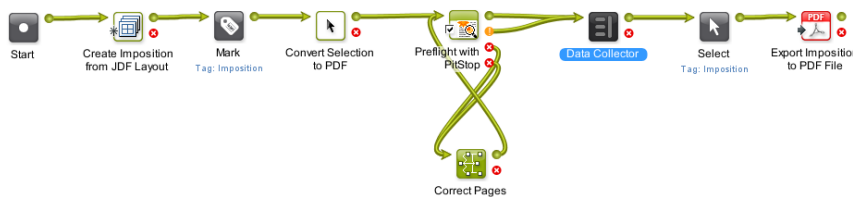**Page Label**: The page labels as they will appear in the output file(s).

Note: You can make use of SmartNames from the categories **Page Information** and **Plate Information** here.

Apply Color Mapping: You can enable this to export the layout file with the same color mapping as defined in the **Plates** view.

Check page status: You can use this option to check if the approval status of the pages is compliant with the **Job setup** settings. The task will fail if it is not compliant.

5. Workflows with Selection and the Convert Selection tasks.

Page, RunList, Layer, PageSlot, Imposition and Plate are abstract file types that do not correspond directly to a file. They can be used as entities in a workflow to route the elements you want to process through the workflow. For example, you can import an Imposition and preflight all the PDF files referred from that Imposition. The workflow below accepts a JDF Layout as its input, selects the PDF files referred in the JDF file and waits until all of them pass the preflight check before producing outputs as sheets to a PDF file. You need to convert the Selection to a PDF before Preflight.

Read more about **Convert Selection** task in *Convert Selection*

5.1 Convert Selection

You can use this task in a workflow to automate file conversion. For example, you can convert some of the layers from an Imposition from the **Pages** view to a PDF format for proofing before proceeding with the next task.

You can convert formats such as

- PDF
- Page
- Runlist
- Layer
- PageSlot
- Imposition
- Plate

into any of these output formats

- PDF
- Page
- Runlist
- Layer
- PageSlot
- Imposition
- Plate

This task is useful in a workflow when you need a certain file format to continue the flow.

You can choose the output format of the task in the **Output Format** field.