EFI Fiery® JobFlow™

Connect Cookbook

FEBRUARY, 2016

Fiery JobFlow Connect

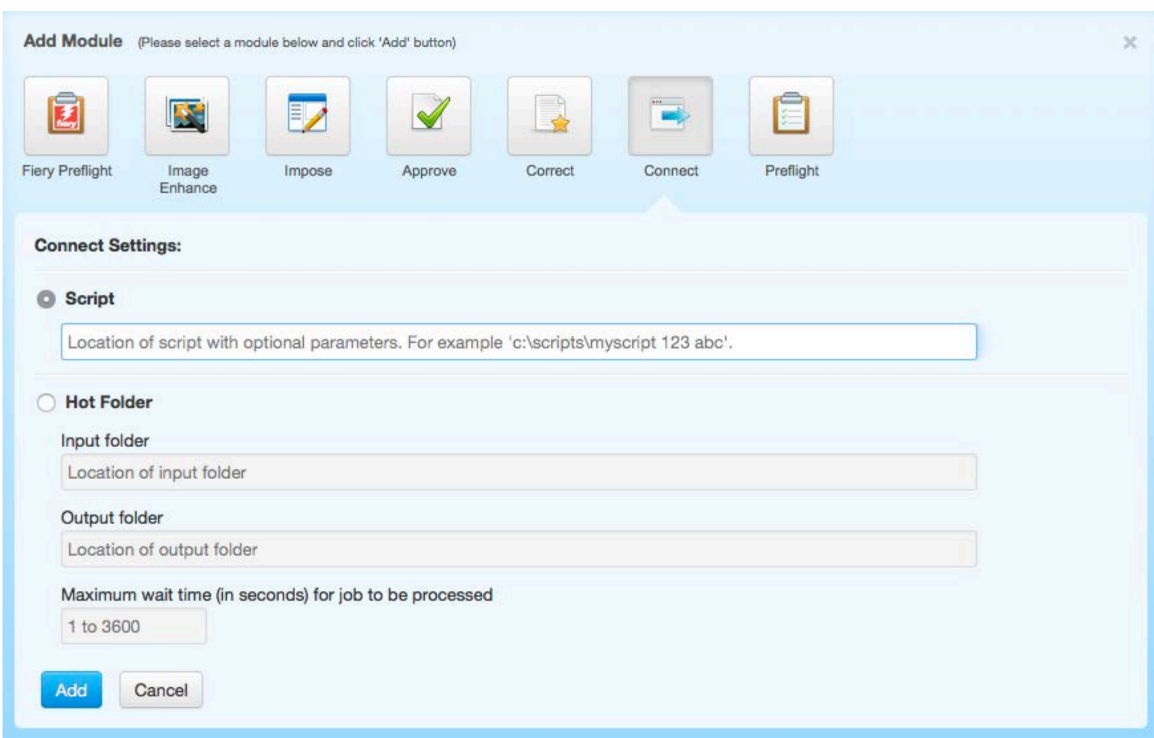
Fiery® JobFlow™ 2.2 introduces a new module called Connect that enables JobFlow users to extend job processing capabilities by connecting to third-party applications using scripting or hot folders. Fiery JobFlow Connect is available in the paid version of Fiery JobFlow.

The Connect module can be used to:

- Convert unsupported native file formats to a format that Fiery JobFlow supports. Add the Connect module to a workflow right after the Input module, but before the Convert module.
- Modify PDF jobs with third-party solutions as a normal step in a Fiery JobFlow workflow. Place the Connect module anywhere in a workflow.

Using hot folders

The first option to submit a job for processing with a third-party solution is by using hot folders. Users need to specify the input and output hot folder location and define a wait time for the processing to occur (up to 60 minutes). JobFlow will move the job to the input location and after the wait time has elapsed, it will pull the job back into the workflow. The wait time allows the third-party application to process the job and upload it in the output folder. Please note the input and output locations defined need to be accessible by the Fiery JobFlow application.



© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

In the following examples the input and output locations are hot folders created on the drive where Fiery JobFlow is running.

c:\hotfolders\input
e:\hotfolders\output

To access shared folders on the network, use the standard Windows® notation for network locations:

\\server\hotfolders\input
\\server\hotfolders\ouput

Using scripting

The second option to submit a job for third-party processing is via scripting. However, scripting requires a certain level of expertise. Users need to be familiar with the basics of scripting and understand concepts such as arguments. **EFI™ does not assume any responsibility for anything that might happen as a result of a poorly written script.**

We have tried to make it as easy as possible for Fiery JobFlow users to implement scripts. To be able to process a job via user-defined scripts, JobFlow will automatically provide the script with the following arguments:

1. **Input location:** a temporary location where Fiery JobFlow will make the file available for processing
2. **Output location:** a temporary location where Fiery JobFlow will wait for the processed file to be copied
3. **Job name:** the job name as shown in Fiery JobFlow
4. **Workflow:** the name of the workflow where the script was initiated
5. **Preflight report:** location of the last preflight report generated in that workflow

Arguments don't have names and the user will need to define in the script the order to interpret the arguments.

Important note: Arguments provided by the user always come first.

In the following example, you can see how you can read values provided by script arguments in to script variables and how to use these variables and their values in the rest of your script.

Most of the scripts shown in this document have a similar structure:

1. Set script variables and values from the arguments provided by the user and Fiery JobFlow. Usually these variables are set by the arguments that are provided to the script by Fiery JobFlow. Note the order of the parameters: because we allow the user to set two values (number of pages and workflow name) the arguments created by Fiery JobFlow come third, fourth, fifth and so forth.
2. Check if the values for the main variables are valid.
   In this case, where we want to split a document in to sections, we don't want values less than 1. We also check if the workflow exists where the sections need to be moved.
3. Check if utilities that we run through a script are installed.
   If we find they are not installed, the script stops and logs the error in a log file that will be written in the folder where the script is running.

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

4. Specify the main procedure that needs to be executed for this script.
   In this case we want to call a tool called pdfbox with the correct arguments.

```
@echo off

:: Script that calls pdfbox (https://pdfbox.apache.org) to split a pdf in to a different workflow
:: Hans Sep, October 16th 2015

:: Fiery JobFlow Arguments
set INPUT=%~3
set OUTPUT=%~4
set JOBNAME=%~n5
set WORKFLOW=%~6
set PREFLIGHT=%~7

:: User Defined Arguments
set PAGES=%~1
:: Note to self: get drive from Fiery Jobflow job
set WorkflowFolder=%~d3\SmartFolders\%~2

:: Static arguments
set BaseScriptPath=%~dp0
set LOG=%BaseScriptPath%\%~n0.log
set PDFBox=%BaseScriptPath%\pdfbox-app-1.8.10.jar

:: Check User Arguments
if %PAGES% lss 1 (
  echo %DATE% %TIME% [ERROR] Pages value not valid: %Limit%. Value needs to be 1 or higher. Exit
script.>>%LOG%
  goto :EOF
)
if not exist %WorkflowFolder% (
  echo %DATE% %TIME% [ERROR] %WorkflowFolder% does not exist. Exit script.>>%LOG%
  goto :EOF
)
::Check if PDFBox exists
if not exist %PDFBox% (
  echo %DATE% %TIME% [ERROR] Can't find PDFBox at %PDFBox%. Download PDFBox from
https://pdfbox.apache.org. Exit script.>>%LOG%
  goto :EOF
)

::Main
echo %DATE% %TIME% [INFO] Start splitting %INPUT% in batches of %PAGES% and send to %WorkflowFolder%
>>%LOG%
call :split
echo %DATE% %TIME% [INFO] Done. Hasta la pasta.>>%LOG%
goto :EOF

:split
:: Call pdfbox to split jobs in to chunks of n pages.
java -jar %PDFBox% PDFSplit -split %PAGES% %WorkflowFolder%\%JOBNAME%.pdf

:: Move original job to output location
copy /Y "%INPUT%" "%OUTPUT%\%JOBNAME%.pdf"
echo %DATE% %TIME% [INFO] %INPUT% has been copied to "%OUTPUT%\%JOBNAME%.pdf" >>%LOG%
```

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

In this example:

- The scripts are located in "c:\scripts"
- The script name is "split.bat"
- The user wants to split a job in to sections of 16 pages
- The workflow is called "booklet"

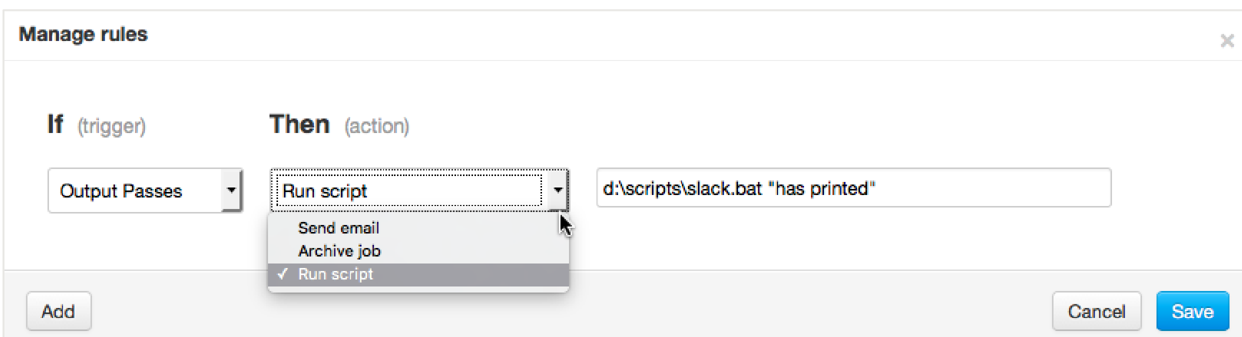
So the correct setting in the Fiery JobFlow Connect module would be as follows:

c:\scripts\split.bat 16 booklet

Fiery JobFlow rules scripts

Fiery JobFlow rules scripts can be used to:

- Archive a Fiery JobFlow job to a location that is not supported by the Fiery JobFlow locations options.
- Send notifications using a third-party solution for users who prefer not to use email notifications.



Rules scripts follow the same conventions as Fiery JobFlow Connect scripts with one exception: rules scripts don't generate output that needs to be processed further in a workflow.

Fiery JobFlow provides the following arguments for rules scripts:

1. **Input location:** a temporary location where Fiery JobFlow will make the file available for processing
2. **Job name:** the job name as shown in JobFlow
3. **Workflow:** the name of the workflow where the script was initiated
4. **Preflight report:** location of the last preflight report generated in that workflow

Important note: Arguments provided by the user always come first.

In the following example, we use a script to do a basic notification through a service called PushBullet: http://www.pushbullet.com.

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

Comments before we get started:

- Install “curl” on the JobFlow server. This is a common tool used to query online services. Download a Windows® installer at http://www.confusedbycode.com/curl/
- Create a PushBullet account. When created users are provided with an “Access Token”, which looks like a string with garbled letters and numbers. Use that token in this script to work.

```
@echo off
::Very, very basic script for push notification via PushBullet (www.pushbullet.com) using curl
::Hans Sep, October 30th 2015

::All static properties
::You need to provide the Access Token which you can find in your PushBullet account
set TOKEN="PUT_YOUR_TOKEN_HERE!"
set URL="https://api.pushbullet.com/v2/pushes"

::User provided arguments
set MESSAGE=%~1

::All arguments provided by Fiery JobFlow
set JOBFULL=%~2
set JOBNAME=%~3
set WORKFLOW=%~4
set PREFLIGHT=%~5

curl --header "Access-Token:%TOKEN%" -X POST %URL% --header "Content-Type: application/json" --data-binary "{\"type\": \"note\", \"title\": \"JobFlow Update\", \"body\": \"%WORKFLOW% %MESSAGE% %JOBNAME%\"}"
```

In this example:

- The scripts are located in "c:\scripts"
- The script name is "pushbullet.bat"
- The message is "[a workflow] has processed [a job]"

So the correct setting in JobFlow rules would be as follows:

c:\scripts\pushbullet.bat "has processed"

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

Supported scripting languages

Fiery JobFlow scripts run on the Fiery JobFlow server so users can use any scripting language that is supported on Windows 7 and above.

Supported languages out of the box

- Windows batch scripting:
  - http://ss64.com/nt/syntax.html
  - http://www.robvanderwoude.com/batchstart.php
  - http://www.incodesystems.com/products/batchfi1.htm
- Microsoft VB script:
  - https://en.wikipedia.org/wiki/VBScript#Examples
  - http://www.robvanderwoude.com/vbstech.php
- Ruby: Fiery JobFlow uses Ruby 1.9 as the underlying programming language. Therefore, all prerequisites for running Ruby scripts are installed on the JobFlow server. Please note that Ruby falls under the "advanced" category, and there are not a lot of PDF manipulation libraries available. Here are two:
  - https://www.ruby-lang.org/en/
  - https://github.com/prawnpdf/prawn

Supported, but not pre-installed

- Python
  - http://www.binpress.com/tutorial/manipulating-pdfs-with-python/167
  - https://github.com/pmaupin/pdfrw/
- Perl
  - http://www.misc-perl-info.com/pdf-perl.html

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

Fiery JobFlow ticket and Fiery JobFlow Connect

Fiery JobFlow ticket

Fiery JobFlow ticket is a feature in Fiery JobFlow Base that allows users to submit jobs with a simple, text-based ticket to define the order of jobs, number of copies per job, and if jobs should be merged or not.

A Fiery JobFlow ticket is a text file with comma-separated values. Currently, Fiery JobFlow supports 2 columns: job location and number of copies. There is no need to add a header or name the columns. Fiery JobFlow always assumes job location is in column 1 and number of copies is in column 2. Comments can be added in the ticket by beginning a line with '//'. Fiery JobFlow will ignore anything in a comment line.

Fiery JobFlow can merge jobs in a ticket by adding an action. An action is preceded by the # character. Currently, JobFlow only supports the #merge command.

Examples

Here is an example of a ticket that retrieves jobs from a local folder or network location and sets the number of copies per job:

// Filename, NumCopies
c:\folder\Bedding Flowers.pdf,10
c:\folder\Direct Sow Flowers.pdf,10
c:\folder\Flowering Bulbs.pdf,5
\\networkdrive\Roses.pdf,20

Here is an example of a ticket that merges the listed jobs and sets the number of copies for the merged job:

// Filename, NumCopies
#merge,10
c:\folder\Bedding Flowers.pdf
c:\folder\Direct Sow Flowers.pdf
c:\folder\Flowering Bulbs.pdf
\\networkdrive\Roses.pdf

Please note that, when merging jobs, Fiery JobFlow automatically adds bookmarks for the first page of every job that is merged. The bookmark title is the same as the job name, minus the file extension. These bookmarks can be used in Fiery JobMaster™ to automatically add tabs.

Here is an example of a ticket that merges the listed jobs and sets the number of copies for the merged job. The jobs and the ticket are submitted as a folder or archive:

// Filename, NumCopies
#merge,10
Bedding Flowers.pdf
Direct Sow Flowers.pdf
Flowering Bulbs.pdf
Roses.pdf

Please note that if users submit a folder or an archive with multiple jobs, all jobs will be merged automatically. With the #merge command you can also define the number of copies for the merged jobs.

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.

Fiery JobFlow ticket and Fiery JobFlow Connect

There is a restriction when using non standard .csv tickets and Fiery JobFlow Connect together.

If a user wants to process a .csv file with the Connect module, they need to change the file extension from .csv to something else (for instance ".ticket"). Then the Connect module will execute the defined actions within the script.

Fiery JobFlow views all .csv files as tickets. If a user submits a regular .csv file that doesn't contain the proper syntax of a JobFlow ticket, it will not be processed. The only way to prevent this, is to change the extension type or the file format.

© Copyright 2016 | ELECTRONICS FOR IMAGING, INC. ALL RIGHTS RESERVED.