# **CLOUDFLOW User Manual**

# Contents

INTRODUCTION	
Copyright	

INTRODUCTION TO CLOUDFLOW	8
What is CLOUDFLOW?	
Database	
CLOUDFLOW Modules	9
Access CLOUDFLOW	

CLOUDFLOW Requirements	10
System requirements for the CLOUDFLOW Application server	
Browser requirements	
Interaction with firewall and anti-virus software	
OpenGL 3.2 support (optional requirement)	11
CLOUDFLOW and virtualization	
Recommendations	
CLOUDFLOW in the network	15
Operating System security context	15
Back up	
SHARE requirements	15

CLOUDFLOW Installation	18
Single CLOUDFLOW server setup	18
Install CLOUDFLOW.	18
Initial setup	19
nucleusd	21
Updating CLOUDFLOW	22
Running CLOUDFLOW over SSL	23
Multiple CLOUDFLOW server setup with fail-over	25
CLOUDFLOW configuration	25
Example of a setup	26
Install extra Work server	29
Backup Best Practices	29
MongoDB Installation	31
Run MongoDB as part of CLOUDFLOW (RECOMMENDED METHOD)	32
Install MongoDB on a separate machine (EXPERT METHOD FOR SPECIAL CASES)	32
Repair MongoDB	
Back up MongoDB	36

CLOUDFLOW Configuration	
DASHBOARD	
LOGGING	
WORKABLES	
FILE STORES	
File store requirements	

Add a File store	
File storage	
OUTPUT DEVICE	
FOLDER MAPPINGS	
WORK SERVERS	
Add a Work server	
Configure a Work server	
Workers overview	
Default Worker setup	
SETTINGS	
Inspect Collections	
CLOUDFLOW Plug-in Suite SETTINGS	
RESOURCES	
PATCHPLANNER	
MAINTENANCE	
LICENSE	

HOME		55
Side b	par	55
User ]	Button	59

UDFLOW WORKSPACE	
Assets	
ASSETS tab	
Asset details	
Pagebuilder	
Activate Pagebuilder	
Create a Pagebuilder page	
Edit an existing Pagebuilder page	
Pagebuilder overview	
Web Page Settings	
Build up the page	
Pagebuilder components	
Translate a page	
Working with Actions	
Working with Page Variables	
Styling	
Use case of PAGEBUILDER	
Workflows	
Building a workflow	
Workflow Editor	
WORKABLES tab	
Advanced Workflow Building	
Nodes overview	
Input	
General Workflow Constructs	
File Handling	
Collaboration	
Assets	
PROOFSCOPE	
PACKZflow	
RIP	
Network	
External Tools	

	Output device	
	Conversions	23
	Share	
	Data	
	Jobs	24
	Mail	
	3D	
	Presets	
KIOSE	Κ	
	Submit a job to a workflow	
	Jackets overview	
	Access KIOSK	
Form 1	Builder	
	How to build a form	
	Overview of the KIOSK parameters	
Users	and Contacts	
	Manage contacts	
	Manage users	2.8
	User Preferences	28
	Admin user and administrators	28
Scopes	S	
200p0	Create a scope	28
	Assign a scope	28
	Guest scope	28
	Restricting access to workflows using scones	28
	Scoping workables	28
	How to filter assets with scopes	28
Variah	les	28
vuriuo	Defining custom variables	29
	Variable specifications	29
	Fynression Builder	29
Appro	vals	29
rippio	Build an approval cycle	29
	Manage approvals	30
	Annroval templates	30

view screened mes	
Compare files	
Approve, reject or delegate files	
PROOFSCOPE shortcuts	
Supported PROOFSCOPE formats	
PROOFSCOPE SDK	
The CLOUDFLOW API	
The PROOFSCOPE component	
1	

PACKZflow	
PACKZflow	

Create a tob	27
Create a sheet	
P	
Screening	
AM screens	
HDS screens	
HXM screens	
Standalone RIP	
Jobs	
Presets	
Curves	
Color Profiles	
Logs.	36
TIFF viewer	
IARE	
Set up a SHARE network	
Define a server as a site	
Add a site	
Create a sync specification	
View the share status	
SHARE best practices	
SHARE best practices	
ATALINK.	
SHARE best practices ATALINK Connect a database	
SHARE best practices ATALINK Connect a database Add a query	
SHARE best practices ATALINK Connect a database Add a query	
SHARE best practices ATALINK Connect a database Add a query DBS	
SHARE best practices ATALINK Connect a database Add a query DBS Job workflow	
SHARE best practices ATALINK Connect a database Add a query DBS Job workflow Job data	
SHARE best practices	
SHARE best practices.         ATALINK.         Connect a database.         Add a query.         DBS.         Job workflow.         Job data.         User Interface.         CONFIGURATION.         Dashboard.         Jobs in KIOSK.         JOBS examples.         Example of JOBS workflows for a label printer.         Use case.         3D.	
SHARE best practices.         ATALINK.         Connect a database.         Add a query.         Add a query.         DBS.         Job workflow.         Job data.         User Interface.         CONFIGURATION.         Dashboard.         Jobs in KIOSK.         JOBS examples.         Example of JOBS workflows for a label printer.         Use case.         3D.	
SHARE best practices	
SHARE best practices	

CFApp	
Install CFApp	394
Use CFApp	
Use case	

MARS	398
What is a MARS app	398
MARS Creator license	
Use MARS without a MARS Creator license	399
Show MARS user interface	
Your list of MARS apps	399
Installing and updating a MARS app	400
Use MARS with a MARS Creator license	400
User roles	400
Types of MARS apps	401
How to	404
MARS app design guidelines	
Scenarios	410
Scenario: installing a public MARS app	411
Scenario: updating a public MARS app	411
Scenario: copying a public MARS app	412
Scenario: creating and uploading a MARS app	413
Scenario: upload a new version of a MARS app to My Apps	414
Scenario: installing a MARS app from My Apps	414
Scenario: publishing a MARS app	415
Scenario: sharing a MARS app	416
Scenario: assigning the co-ownership of a MARS app	416
Scenario: installing and updating a MARS app at a customer	418

OUDFLOW Plug-in Suite	
Installation	
Illustrator Plug-in	
Connect to the CLOUDFLOW server	
File menu	
Window menu	
PACKZ Plug-in	
Enable CLOUDFLOW Plug-in Suite in PACKZ	
Use CLOUDFLOW Plug-in Suite in PACKZ	
Desktop Service	
Configuration	
Functionalities	

COCKPIT	
Install COCKPIT	
Initiate COCKPIT	
Use COCKPIT	
UPLOAD	
DELIVERIES	
ITEMS	
JOBS	
TOOLS	

ADMIN	
Template and customizations	

TECTONICS	
Prerequisites	
Configuration	
User interface	
Placement	
Queues	
Cutoffs	
Archive	
Statistics	
Settings	

INTELLIGENT FLEXO	
Description	
Intelligent Flexo Process	
Prerequisites	
Configuration	
Intelligent Flexo Node	
Building a Workflow	

VDP.	

)P	
Prerequisites	
Configuration	
User interface	
Job Station	
Job management	
Create a Job	

APPENDIX	
PROOFSCOPE Quick Start	
EN version	
Data Structure of HelioDisk Job Details	
Functions in the Script node	
Using custom objects in the database	
Output File Path Generation	

GLOSSARY	
Regular expressions	
Index	

# INTRODUCTION

# Copyright

Copyright<sup> $\infty$ </sup> 2021, Hybrid Software NV. All rights reserved. This document may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without written approval from Hybrid Software NV.

# INTRODUCTION TO CLOUDFLOW

# What is CLOUDFLOW?

CLOUDFLOW is a modular server software system for running automated graphics production workflows in a private computing cloud. It provides various services to manage and automate data for your graphic arts business.

CLOUDFLOW is a web-based application platform and can be installed on one or more servers. All interfaces are developed in HTML5, and the system can be controlled from a web browser or computing tablet.

## Database

CLOUDFLOW uses MongoDB as a database to store all the assets. It can store locally, on one or more servers, or in a cloud. It has support for password protected MongoDB databases.

## **Database collections**

MongoDB stores records in collections as JSON objects.

Example of a record in JSON format:

```
{
   "id" : "123",
   "url" : "Filestores/DemoResources/demo files input/10_SnR_FoldingCarton/
XML 7 designs in SnR.xml",
   "path" : [
        "Filestores",
        "DemoResources",
        "demo files input",
        "10_SnR_FoldingCarton"]
}
```

## A brief explanation on JSON syntax

• JSON objects are surrounded by curly brackets {}.

• The two main parts in a JSON object are keys and values. Together they form key/value pairs.

## Example

"id" : "123"

- "id" = the key
- "123" = the value
- Keys must be strings and values must be one of the following data types:
  - a string: { "id" : "123" }
  - a number: { "amount" : 30 }
  - a JSON object:

```
{
    "file":{ "id" : "123", "amount":30 }
}
• an array:
    {
    "path" : [ "Filestores", "DemoResources", "demo files input",
        "10_SnR_FoldingCarton" ]
    }
• a boolean: { "checkbox" : true }
```

- **null**: { "sub" : null }
- Keys and values are separated by a **colon**.
- Each key/value pair is separated by a **comma**.

# **CLOUDFLOW Modules**

CLOUDFLOW consists of different modules:

## **Basic module**

CLOUDFLOW Workspace

## **Optional modules**<sup>1</sup>

- PROOFSCOPE
- PACKZflow
- PRINTPLANNER
- PATCHPLANNER
- RIP
- CLOUDFLOW SHARE
- FACELIFT
- DATALINK
- JOBS
- 3D
- MARS
- CLOUDFLOW Plug-in Suite

<sup>&</sup>lt;sup>1</sup> Contact your local sales representative for more information.

# Access CLOUDFLOW

By default, CLOUDFLOW listens to port number 9090. If you open CLOUDFLOW on the local host, you can access CLOUDFLOW on http://localhost:9090. If you open CLOUDFLOW on another computer, you have to enter the IP address of the host where CLOUDFLOW is installed, for example http://127.0.0.1:9090.

# **CLOUDFLOW Requirements**

# System requirements for the CLOUDFLOW Application server

These are the recommended system requirements for the CLOUDFLOW Application server.

**Important:** The server instances that are provisioned for CLOUDFLOW and MongoDB should only run the CLOUDFLOW and/or MongoDB software. No other software should be installed and running on those systems, as this will potentially hurt performance and functionality of the system.

## System

=

- Windows 2012 Server 64-bit system or higher.
- Mac OS X 10.12 or higher.

**Note:** If you are running PACKZFLOW 6.0 on Mac, the minimum system requirement is MAC OS X 10.14.

## Hard disk space

Minimum of 500 GB of disk space though multiples are not uncommon depending on projected activity.

## **CPU** and memory

CPU and memory depend on the modules.

The following modules are recommended to have 8 Threads and 16 GB of memory each:

- WORKSPACE
- PROOFSCOPE
- PACKZFLOW
- RIP

The following module is recommended to have 8 Threads and 32 GB of memory:

VDP Execute

The calculation is cumulative.

## Two examples of a hardware configuration

Setup	Required Threads	Required Memory
Server with 1 WORKSPACE + 1 PROOFSCOPE	$2 \ge 8$ Threads = <b>16 Threads</b>	2 x 16 GB = 32 GB
Server with 1 WORKSPACE + 2 PACKZflow + 1 RIP	4 x 8 Threads = <b>32 Threads</b>	4 x 16 GB = 64 GB

## Scalability

These system requirements will be sufficient for the majority of all CLOUDFLOW installations. Elaborate CLOUDFLOW set-up featuring extra modules will require supplementary system resources.

Additionally, actual system requirements depend on the project's workload, worker configuration, the user's system configuration, workflows and subsequent expected workload throughput. If these conditions are taken into account CLOUDFLOW can be scaled to match operational workload.

## **Browser requirements**

The browser to control the CLOUDFLOW user interface on the CLOUDFLOW client side needs to comply with specific requirements.

CLOUDFLOW interfaces are developed in HTML5, and can be controlled from a web browser or computing tablet. It is recommended to download the newest version of your browser for the best experience.

CLOUDFLOW supports the following browsers:



- Chrome 51 and later.
- Firefox 46 and later.
- Microsoft Edge 38 and later.



- Chrome 51 and later.
- Firefox 46 and later.
- Safari 9 or later.

# Interaction with firewall and anti-virus software

The CLOUDFLOW database binds to the IP address of the machine, and uses default TCP port 9090.

If **firewall** software or **anti-virus** software is installed on the same machine, make sure that this software is configured appropriately in order not to interfere with the TCP communications between CLOUDFLOW and MongoDB.

# **OpenGL 3.2 support (optional requirement)**

If you are running the **3D option** in CLOUDFLOW, your server needs to be equipped with a 3D capable video card which has at least 1.5 GB of VRAM available and has a driver installed with at least the OpenGL 3.2 API.

## **OpenGL and Windows Service incompatibility**

Windows doesn't allow applications that are started via the service manager to access OpenGL. If 3D is to be used in CLOUDFLOW running on Windows, you can to start the CLOUDFLOW software from a computer start-up script.

# **CLOUDFLOW** and virtualization

For ease of IT management, it is recommended to run CLOUDFLOW and MongoDB on virtual machines.

## Cloud vs. physical setup

CLOUDFLOW functions perfectly whether hosted on physical servers or on cloud based servers. Both setups support all of CLOUDFLOW's components and infrastructures. When deciding for either one option, the considerations to be made partly relate to the following general advantages and drawbacks:

	Physical	Cloud
Server	<ol> <li>1 DEDICATED machine</li> <li>user dependent</li> </ol>	<ul> <li>space on SEVERAL shared machines</li> <li>hardware independent</li> <li>automatable</li> </ul>
Performance	<ul> <li>backups inhibit performance</li> <li>data relocation inhibits performance</li> <li>hardware not easily scalable: RAM, CPU, storage,</li> <li>each user account benefits from all available hardware</li> <li>Specific configuration and tweaks optimize flexibility and performance</li> </ul>	<ul> <li>easy and quick backup</li> <li>easy and quick relocation</li> <li>1 click (and automatable) scalable hardware: RAM, CPU, storage, </li> <li>user accounts share hardware resources</li> <li>General configuration, standard performance</li> </ul>
Location	On-site. Impact on maintenance, risk, responsibility Off site. No impact. Maintenance risk and responsibility covered by provider contract	
Maintenance	Downtimes inevitable	Limited downtime
Access	Operators on-site, no internet required only for cloud Operators can access from any location that has internet availab	
Access	No internet access required Internet access required	

An additional consideration, however, is the concept of **Data Locality**. Data Locality is the tendency of a processor to reuse similar memory locations repetitively. Such reuse favours memory locations that are neighbours in space and/ or time. Conversely, similar datasets that are accessed over time intervals or spatial distances that are too long will lose the benefit of reuse. As a consequence, CLOUDFLOW's data must be kept in closest proximity to its engines, whether in the cloud, locally, or temporally.

This means that sufficient consideration must be given to make the architecture of the CLOUDFLOW setup withstand expected loads and work-flows. Elements to consider as well: network reliability, cable quality, number of operators, mean data load, downtime proneness, etc. where files are stored and where memory see whiteboard

In the case of CF the basic minimal requirement is a user downloading and uploading files from a DB, processing them CF and then re-uploading them (via CF) to the DB. These files are typically large and slow in data traffic.



Consider the image above where:

- 1. On the left we have the blue work-flow, a local CF setup with user 1 (U1) interacting with CF1 that in turn interacts with a DB1. As long as user and DB are local, there is no problem.
- 2. In the orange work-flow, U1 is local but DB1 is not local (remote or virtual) and so both the user and CF1 will be interacting with the remote DB1 and that will require time.

One way to solve this is to increase spatial data locality and deploy another CF instance, CF2, in physical proximity of the remote DB1. That way at least CF2 and DB2 are 'neighbours'.

User1 won't experience any drawbacks as his direct partner is still CF1 and even though CF1 connects remotely to CF2 it does so with compressed data packages, which synchronize much more quickly. The technology to share and synchronize information between separate and independent CLOUDFLOW servers across the globe is described in the chapter SHARE on page 363.

In more realistic setups requirements are stricter as those will see much more traffic, from many more users.

Assume U2 and U3 want to view and validate U1's file in Proofscope. Both U2 and U3 are in another physical location. To them all setup elements of solution 2 above are (too) remote: U1, DB1, CF1 and CF2, so that transferring and downloading a file to them directly draws on the system.

However, U2 and U3 do not need the data as soon as U1 has processed the data. The accepted work-flow allows for ample time for the system to send them a notification mail with a link for them to download their file to validate. Or even a notification mail with either a full size file or a compressed one.

CLOUDFLOW setups allow for ample possibilities but the examples above demonstrate that customers' individual requirements must determine the way the work-flows are conceived so that the architectural landscape can be optimized.

To conclude:

- As long as User+DB+CF are local, the setup is optimal.
- As soon as either CF or DB are not local (cloud, virtual, remote), then the other needs to be deployed there too.
- First work-flows, then setup.

## **Memory requirements**

CLOUDFLOW runs several different sub processes (metadata extraction, thumbnail calculation, pre-rendering for **PROOFSCOPE**, workflow module, etc...). All these components work together, and each component needs CPU power and memory. Some components (for example the (pre-)RIPping or the pre-press workflow) need sufficient memory to function properly.

On a production CLOUDFLOW system, the CPU often uses close to 100% of its cores and a big chunk of the available system memory. An operating system can handle overdrawing CPU, so if there are more components that want to use the system than the system can handle, performance degrades gracefully. Unfortunately, this is not the case with overdrawing memory. When the total memory consumption exceeds the provided memory, performance drops. This results in an unresponsive and sometimes freezing system.

Therefore, the system needs sufficient memory. See System requirements for the CLOUDFLOW Application server on page 10 for the recommended minimum requirements.

#### Combining systems on one host

Adding a virtualization layer to the hardware causes a performance hit.

Depending on the technology this can be small (5%) or large (30%). One of the drivers of virtualization is making better use of hardware, by combining multiple guest systems on one host.

However, since CLOUDFLOW is designed to extract all the performance it can get from the underlying system (by making heavy use of multi-threading and multi-processing), it is at odds with other virtual guests competing for the same performance of the underlying host.

Only when the underlying hardware is very highly spec'ed, this argument becomes less of an issue (for example 16core servers with 256GB RAM).

## **3D and OpenGL limitations**

It is not possible to use CLOUDFLOW's 3D capabilities when running on top of a virtualization software (like VMWare, Hyper-V, ...).

CLOUDFLOW's 3D rendering technology uses OpenGL 3.2 or higher. This is, to our current knowledge and field trials, not supported by current visualization technologies.

If CLOUDFLOW 3D technology is used, it is recommended to run the software non-virtualized on a hardware platform with 3D capabilities and a graphics driver that offers the Open GL 3.2 API for the installed operating system.

## Recommendations

These are the recommendations if you run CLOUDFLOW and/or MongoDB in virtualized containers:

#### Use high performance hardware

Make sure the underlying hardware platform is very powerful. CPU is important, but memory is more important. Add lots of memory in the machine (64GB).

## Allocate lots of memory

Add sufficient memory to the CLOUDFLOW instance. See System requirements for the CLOUDFLOW Application server on page 10 for the requirements.

#### Do not run other images on the same machine

Only run a single virtual container with CLOUDFLOW per hardware server, and allocate all resources to that CLOUDFLOW instance.

# **CLOUDFLOW** in the network

These are the requirements for CLOUDFLOW in the network:

- The CLOUDFLOW server software should be able to reach the database server over IP.
- If the CLOUDFLOW server needs to index shares, these shares should be accessible for the CLOUDFLOW server.
- If CLOUDFLOW runs as a service on Windows, the shares should be made accessible to CLOUDFLOW running as a service.

# **Operating System security context**

Many problems during installation are a result of incorrect setup of the operating system permissions of the various parts of the software. To operate correctly, the CLOUDFLOW server software needs to have access to:

- its binary files (installation files)
- the File stores (customer data)

**Note:** CLOUDFLOW is installed as a service and will run under a specific user account. Make sure the user account used for running the service can access (read and write) to the configured File stores.

## Back up

=

How to back up MongoDB in Cloudflow.

CLOUDFLOW does not provide a feature to (automatically) back up MongoDB or your File stores, so you need to make sure to regularly back these up individually.

Note:

More details on how to back up your MongoDB installation can be found :

- in the Backup Best Practices on page 29 topic of the CLOUDFLOW manual.
- on the MongoDB website.

# **SHARE requirements**

This chapter gives an overview of the requirements you need to take into account for a good performance of **CLOUDFLOW SHARE**.

**CLOUDFLOW SHARE** is a technology to share and synchronize information between separate and independent CLOUDFLOW servers across the globe. See CLOUDFLOW SHARE for more information.

If your system does not meet these requirements, a good performance of CLOUDFLOW SHARE is not guaranteed and possible issues and problems will arise.

## Network

These are the network requirements.

- The **bidirectional interconnection speed** between the different sites needs to be at least 10 Mbit/s (megabit per second).
- The network latency needs to be lower than 1 second. You can test this by pinging.

• The **bandwidth** depends on the amount of data you want to transmit. This can be calculated as by estimating the amount of data that will be interchanged between the sites during an acceptable time interval that is needed for the transmission of the data from one site to another (for example 1 hour).

## Example

In 1 hour, 20 job folders are exchanged. The folders contain of 200 MB of data, which makes a total of 4 GB. this needs 10 Mbit/s, dedicated for CLOUDFLOW SHARE.

• A direct and reliable connection between the CF servers needs to exist.

## File store

• The average consistent **indexing speed** of the complete File store you want to use on CLOUDFLOW SHARE needs to be lower than 5 minutes. If it consistently takes longer than 5 minutes to index the File store, all CLOUDFLOW SHARE transactions will be slowed down.

What can you do to improve the indexing speed?

- Split up the File store to sub-directories.
- Decrease the volume of files that are on the File store.
- Speed up the storage (for example move the storage to CLOUDFLOW or vice versa).

Or you can try a combination of the above measures.

- The file system needs to read and manage the **metadata** (such as modification date) correctly.
- The resource forks and alternate data streams need to be manages identically on both sites.
- The operating system and storage system need to be identical on both sites.

## **CLOUDFLOW SHARE Setup**

• Make sure to work **transactional** and to **restrict the activity** on both sites on the same time of the synchronized CLOUDFLOW SHARE folders. Limit the concurrent file transactions on these folders. A high concurrent folder

activity and too many file transactions imply many decisions that CLOUDFLOW SHARE needs to take, which may cause conflicts and unexpected results.

You can use CLOUDFLOW nodes to set up and manage a transactional workflow.

## Example of a transactional workflow

This is an example of a transactional workflow with check-out and check-in.

## Setup

- A central CLOUDFLOW system where all production files are stored.
- Two production sites.
- Two File stores on each site:
  - RO (Read Only) with read permissions only and where the operators save the files when they are finished. This File store is synchronized with the central CLOUDFLOW site through a synchronization mapping.
  - RW (Read and Write) with read and write permissions and where the operators work on the files.

## Workflow

When an operator wants to work on a file, the operator checks out the file to the RW File store in the production site. When ready, the operator copies the file to the RO File store in the same production site. This File store is synchronized with the central CLOUDFLOW site through a synchronization mapping.

## Workflow nodes in CLOUDFLOW



## Check out

- Check Out: starts the check-out workflow.
- Add Sync Spec Mapping: makes sure the RO File store is synchronized with the central COUDFLOW.
- Wait For sync: makes sure the workflows waits until the files are synchronized.
- **Copy Folder**: copies the files from the RO File store to the RW File store so the operators have access to the files and can work on them.
- Notify: script that sends out a notification of the action.

## Check in

- Check In: starts the check-in workflow.
- Copy Folder: copies the files from the RW File store to the RO File store.
- Wait For sync: makes sure the workflows waits until the files are synchronized.
- Notify: script that sends out a notification of the action.
- The maximum amount of **files** in a mapping should be **500**. The maximum amount of **mappings** in a sync spec should be **25**. This makes a maximum total of **12.500 files in a syncspec**. It is also important to keep the amount of mappings and the amount of syncspecs balanced. Treat a sync spec as a job folder, with subfolders (TBA).
- Disable obsolete sync specs.

## Things not to do

Avoid the following:

- Full File server syncing between sites.
- Full active WIP (Work in Progress) folder syncing.
- Overlap of mappings.

# **CLOUDFLOW** Installation

CLOUDFLOW is a server software product with a database in the back end. It can be deployed on one or multiple servers. To install it, you need to install:

- 1. The database, called MongoDB.
- 2. CLOUDFLOW.

The following chapters will guide you through the installation process of both MongoDB and CLOUDFLOW.

See also the comprehensive CLOUDFLOW User Manual for further details.

# Single CLOUDFLOW server setup

The CLOUDFLOW server runs on Windows Server or on **W**.

The system features an embedded web server and uses multi-processing and threading. The system uses a fair amount of memory during tasks like pre-rendering for **PROOFSCOPE** soft proofing, **PACKZflow** processing, etc... When the web server runs on the same machine as the rendering, the user running the CLOUDSERVER service should have administration rights on the machine to allow elevating the web server priority.

## Install CLOUDFLOW

To install CLOUDFLOW, follow these steps:

- 1. Make sure that there is no other instance of CLOUDFLOW running on the system where you want to install it.
- 2. Log in to the Data Center and download CLOUDFLOW.
- 3. Extract the software to the correct location.

The recommended installation location for CLOUDFLOW is:

	ú
C:\Program Files\Cloudflow\	/Applications/Cloudflow/

## **4.** Start the install routine:

2	ú
Run the Command Prompt as Administrator and type:	Open the Terminal application and type:
C: cd "\Program Files\Cloudflow" .\nucleusd.exeinstall	cd /Applications/Cloudflow sudo ./nucleusdinstall

**5.** Create the PP\_FILE\_STORE folder.

		Ú	
•	previous location:	•	previous location:
	C:\ProgramData\NiXPS\Filestore\		/Users/Shared/NiXPS
•	recommended location:	•	recommended location:
	C:\work\CLOUDFLOW\PP_FILE_STORE\		C:\work\CLOUDFLOW\PP_FILE_STORE\

This is the folder that stores all files submitted and generated CLOUDFLOW and the MIYAKOHSI app. This location will need to be referred to in Step 2. Filestore on page 20 during Initial configuration on page 20 of CLOUDFLOW.

See nucleusd on page 21 for detailed information about nucleusd and how to manage it.

## First installation on Catalina blocked

When first installing CLOUDFLOW on with Catalina, the installation of nucleusd is blocked for security reasons. The following message appears:

"nucleusd" cannot be opened because the developer cannot be verified. macOS cannot verify that this app is free from malware.

To install nucluesd, take the following steps:

- 1. Select Cancel.
- 2. Go to System Preferences > Security & Privacy > General. The message "nucleusd" was blocked from use because it is not from an identified developer is displayed.
- 3. Select Allow Anyway.
- 4. If you continue the installation of nucluesd, the message *macOS cannot verify the developer of "nucleusd". Are you sure you want to open it?*.
- 5. Select Open.

## **Initial setup**

When you have installed CLOUDFLOW and you use it for the first time, you have to install your license and define your initial CLOUDFLOW configuration.

## License

If you start CLOUDFLOW for the first time, you have to install your license.

You are automatically directed to the Setup Page and prompted for your license.

🗞 Se	tup Page X	+	-		:
$\leftrightarrow$ $\rightarrow$	C i localhost:9090		☆	00	)
Es	Setup Page			20.4	
	License				
	Customer Code	US-ABCX7			
	Serial	123456			
		Download and install license Upload license file			

Enter the Customer Code and the Serial, and click Download and install license.

After the license is installed, you will be directed to the **Initial Configuration** page. See Initial configuration on page 20.

## Initial configuration

On the Initial Configuration page, you have to specify user and source file parameters.

Setup Page	
Site	
Site Name	DEVELOPMENT •
Cloudflow Adminis	trator
Login	admin
Password	
Check Password	
Filestore	
Filestore Name	PP_FILE_STORE
Filestore Location	C\work\CLOUDFLOW\PP_FILE_STORE\
Optional SMB Login	Username Password
	File store settings ok
Test File	Optional file to check Check

Define the following parameters:

## 1. Cloudflow Administrator

Choose and confirm a password.

- Login: if predefined cannot be modified.
- Password: the initial password for the administrator account for this CLOUDFLOW setup.
- Check Password: repeat the password.

## 2. Filestore

Specify Filestore parameters.

• Filestore Name: the initial name of the File Store. By default, this is PP\_FILE\_STORE.

• Filestore Location: Enter the path of the folder created and defined in the stage: Install CLOUDFLOW on page 18

The location points to a folder that will be indexed recursively. This location will also be used to store uploaded files.

- **Optional SMB Login**: the credentials to log in to an SMB file share. These fields are optional.
- **Test file**: upload any file/folder to the Filestore and specify its name in this field.

**Note:** For configuration to be successful the Filestore cannot be empty. Make sure to add an (empty) file or folder to the File Store before you check it.

• Check: Click Check to verify whether the configured parameters are valid.

## 3. Save configuration

After you have completed the necessary fields, click Save to save your configuration.

Next you will be directed to the CLOUDFLOW log in page.

## nucleusd

nucleusd is the master process in CLOUDFLOW which distributes the work to all the Workers.

You can manage nucleusd via:

=

Command Prompt	🗳 Terminal
----------------	------------

You can run the following commands:

- ./nucleusd --install: with this command you can install CLOUDFLOW. Additional options:
  - -i SERVER\_ID: with this command you can launch the portal daemon with another ID (for example for a secondary node). Default: **PP\_WORK\_SERVER**.
  - -d DATABASE\_IP: with this command you can connect to MongoDB installed at ip db\_ip. Default: 127.0.0.1.
    - **Note:** use -d DATABASE IP: PORT if MongoDB is not installed on the default port.
      - **Note:** If MongoDB is protected with a user name and password, you need to add the following command:

<b>**</b>	<b></b>
-d	-d 192.168.x.x\;authUser:USER
192.168.x.x;authUser:USER;authPass:PASS	\;authPass:PASS

- -p PORT: with this command you can connect to the port of the web server.
- -s: with this command you can silence the web server.
- -u: with this command you can run CLOUDFLOW as another user.

-u	[domain\]username[:password]	
Where domain is the Windows domain (NETBIOS name). Omit this for local users. If no password is specified, nucleusd asks for the password.		

<b></b>		
-u username[:groupname]		
Examples		
-u cfuser		

Examples	-u cfuser:staff
-u Administrator -u MyDomain\CFUser -u MyDomain\CFUser:secret	

- • --launchmongo: with this command you can start the internal Mongo server instead of a standalone one.
  - --ssl cert+key.pem: with this command you can specify the certificate and private key in case you have configured CLOUDFLOW to run over SSL. The data needs to be in the file specified. See Running CLOUDFLOW over SSL on page 23 for more information.
- ./nucleusd --uninstall: with this command you can uninstall CLOUDFLOW.
- ./nucleusd --start: with this command you can start CLOUDFLOW.
- ./nucleusd --stop: with this command you can stop CLOUDFLOW.
- ./nucleusd --version: with this command you can retrieve the CLOUDFLOW version.
- ./nucleusd --status: with this command you can retrieve the status of CLOUDFLOW.

## Example

nucleusd --install -s -d 192.186.1.17 -i PP\_SECONDARY\_NODE will launch a portal that connects to the database at 192.186.1.17, without starting a web server and by using the node-id **PP\_SECONDARY\_NODE**.

## Updating CLOUDFLOW

If you want to upgrade an existing CLOUDFLOW installation it is recommended to first uninstall the existing installation.

## 1. Uninstalling CLOUDFLOW

1. Select the appropriate method:

Open the <b>Command Prompt</b> as Administrator and type:	Open the Terminal application and type:
cd C:\Program Files \Cloudflow (browse to the folder where the old nucleusd is located) nucleusd.exeuninstall	<pre>cd /Applications/Cloudflow (browse    to the folder where the new    nucleusd is located) sudo ./nucleusduninstall</pre>

## 1. Updating CLOUDFLOW

Every time there's a new release of CLOUDFLOW you will be informed by e-mail. To install it, follow these steps:

- 1. Download the updated CLOUDFLOW version.
- 2. Uninstall CLOUDFLOW, as per the method above under #unique\_29/ unique\_29\_Connect\_42\_sectiondiv\_avf\_zpz\_p4b on page 22.
- 3. Select the appropriate method:

	Ú.
Open the <b>Command Prompt</b> as Administrator and type:	Open the Terminal application and type:
cd C:\Program Files\Cloudflow (go to the folder where the new nucleusd is located)	cd /Applications/Cloudflow (browse to the folder where the new nucleusd is located)

.\nucleusd.exeinstall	sudo ./nucleusinstall

Note: Add -d (*IP address*) in case mongoDB is installed on another machine.

Note: In case of different installation path, you need to adjust the instructions as required.

4. Start CLOUDFLOW

=

## Running CLOUDFLOW over SSL

It is possible to setup CLOUDFLOW to run over SSL.

To do so, follow these steps:

## 1. Obtain a certificate

An SSL server needs a certificate. It's important that the certificate is converted to the **pem** format. For example PFX or PKCS#12 formatted certificates (typically for Microsoft IIS) need to be converted to a PEM formatted certificate (for Apache).

The PEM file is normally supplied by the SSL provider. You don't have to generate the PEM file yourself.

The PEM file needs to contain the certificate and the key; the key cannot contain a password. For testing you can generate a PEM file with the following command on OSX:

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 30 - nodes
```

This will give you a cert.pem and key.pem, you need to append the key to the cert:

cat key.pem >> cert.pem

The end result is a single resulting pem file: cert.pem

The PEM data has the following format:

```
-----BEGIN something----
base64-encoded data
-----END something-----
```

It is possible that the PEM is delivered in separate parts. In that case you need to combine them using a text editor in the following order:

- Your private key.
- Your certificate.
- Any intermediate certificates (only if they are supplied by the SSL provider).

**Important:** The certificate needs to be present before the intermediates.

**Important:** The certificate will only validate if you use the correct URL in the browser. If you use https://localhost, you will get an error.

## Example

This is the content of such a self-signed pem file:

```
----BEGIN CERTIFICATE----
MIIDtTCCAp2gAwIBAgIJAN07dYJz/tVpMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
BAYTAkJFMRMwEQYDVQQIEwpTb211LVN0YXR1MSEwHwYDVQQKExhJbnR1cm51dCBX
```

```
aWRnaXRzIFB0eSBMdGQwHhcNMTUxMjAzMTQ0ODQ4WhcNMTYwMTAyMTQ0ODQ4WjBF
MQswCQYDVQQGEwJCRTETMBEGA1UECBMKU29tZS1TdGF0ZTEhMB8GA1UEChMYSW50
ZXJuZXQgV21kZ210cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEA36nuRpY2rG89T3Ygfnwn9k6pMerRDbb+MhZrT2SuQkp8bQv9hMhERZnP
rkvY6K7tQE5BkNYvlzjkr1+XAeDE1172I7/8VmrOv4wniatJWmPjU+MwZONJbIfk
DuWoiwGiN5iNARSwkKpEocQh1zihdDgwge/swLT2vqwE121jg+zgu9jv9Rkhn6Ak
nD8MPYnzsIlllPG2Rb7I/J5N6XQ9IwcZZCCkY+hdEx99w0NsVf4CJlN/Wj0s3Pjh
BCBHS0ihVW+Rwyd2nMlhmJCnXFx15UCncxfToy2Jrln/P0g9v3wIy0TGJ1pWtCOv
ZpdaUcY0MfffssJf+A7WF0u2QkqQgwIDAQABo4GnMIGkMB0GA1UdDgQWBBS/tn3p
60jmZcs4NukRY7vGCYAzYDB1BgNVHSMEbjBsgBS/tn3p60jmZcs4NukRY7vGCYAz
YKFJpEcwRTELMAkGA1UEBhMCQkUxEzARBgNVBAgTC1NvbWUtU3RhdGUxITAfBgNV
BAoTGEludGVybmV0IFdpZGdpdHMgUHR5IEx0ZIIJAN07dYJz/tVpMAwGA1UdEwQF
MAMBAf8wDQYJKoZIhvcNAQEFBQADggEBAJGVBeQ7U0tEBM//C/zwq1qrUmsFvUPK
7zcQn1zKLIbdr/TQ/OcGwEGzte3QEyOquekX1TRWYLdvQ6/rbYA8fdBoHujTNTLL
1FYDaYvTOfjSPf+CvnN69VN2x3t2yjSyyK4JjyDNT+S/v7lrlDCQOPHCnV2oQBH5
zKcgEKMwKoCmnmh9KB0+AbUe+/ZCoqEu93uDX+sQZKZy6ev4oA7pKaIDayEcEqsA
95CWKVXFWwgGd00LeUtEFpl0SCeB8v1cfI0xBENhP2zfeOZ0uiMrHwfsfyfBgrAc
UNFE/i50UYmpD0mXv/MFV3M7yQZrIVBzyBlxRiQ3XODwZMqi+7zn5dE=
----END CERTIFICATE----
----BEGIN RSA PRIVATE KEY----
MIIEpQIBAAKCAQEA36nuRpY2rG89T3Ygfnwn9k6pMerRDbb+MhZrT2SuQkp8bQv9
hMhERZnPrkvY6K7tQE5BkNYvlzjkr1+XAeDE1172I7/8VmrOv4wniatJWmPjU+Mw
ZONJbIfkDuWoiwGiN5iNARSwkKpEocQh1zihdDgwge/swLT2vgwEl21jg+zgu9jv
9Rkhn6AknD8MPYnzsIll1PG2Rb7I/J5N6XQ9IwcZZCCkY+hdEx99w0NsVf4CJlN/
Wj0s3PjhBCBHS0ihVW+Rwyd2nMlhmJCnXFx15UCncxfToy2Jrln/P0q9v3wIy0TG
J1pWtCOvZpdaUcY0MfffssJf+A7WF0u2QkqQgwIDAQABAoIBAQDKw08hyfZSJp07
tWqLqV1wqSoz+Bv/BM42daBd3nUh3wggimgNwMYzGhXseRFvDXRBgS9qrt+BhK+6
Uzs/FpUacBlNmzKS3EwD1HfbEw1yBW8EksVCD9B1tFKHNqWtLZPyNjZMrobI/bEt
u/C5e7rQ8kb245qJrwKnIqUe5Qizg4zKua2/ehsci21xOklaanAQyyaCF0b8Pu9q
MZ4AghW6F5ta/mRiiXyiaIe6j/9XZRStbyCNSGNl3gl0EpgdysFZLT/dM14cCZRw
N1+HeYJ5NQsRbjsucwPU00jzSuDOeV6h4e2P5cmCmoh01XQ75YW525fVGTe496/b
K9ynovpRAoGBAPY3gs2Dg6RMwip+eSZMDIkiV7z4M35cTNW3bU/A5v/4H8ydhRKG
Me5yTMy0eYWd5mrinWlsjqqooyq90ZY9yCM3j2rbwS+g1F23Rb7aLdpyggPOMvdD
HuMC81R36Y8zYZWNx9RVQyos7H+2QYrp65FXUqsyLZSw+4yuc75goTGtAoGBAOiN
A01z5yBFgbqqzOksDPcppnZjrwUlFSKR71BbakzcP8ru6mBMq8IGmyouHdd8Js9i
GXX10iCNSNB3ge2gpt912ehyXozCfAsV7IzMxpHsEnLoJwlS/cr6a1bhGqrjK8a+
dIby2cOZ9a8wPr6jQDpI9axk3GqqOA55td4Hh/DvAoGBALKoAO+i2Cil0dYHw1y4
nff7xXJHHwY3b0E8QcM3E+b0Yq6U1fRlD2IX6qY3hl4f/wvbt6DBtpu6lfrzJxAP
cGOtcxF0t68Zc860p5Uet1rk79ZXDsBAIe9TOqd0ozLXc52TLUazP3Mq+dKJqvFj
Xx0tlb07Qbu82Cpyq2XGs/BZAoGAPq3DNQkIQqp5TNa8ZBx7YiLXuxEjGz/jcm3i
zcXV4OF8UDYU9d+0a5dcOwIzCOFtlHfcZO6zvBJYjdeoLWQflhbdpMkussG4rUQ7
fbezzaDTWCU6YEj+HrdgNwp1JZoVMnwi7DyIdTAKCC34u2lE5nDDxrNotKf6c+xa
GH3vJgECgYEAg2M/Ogj+5x5rT7N+PQBzQXHBo/RKyYX0r9q6BJbiF/4KtJXEd8I0
CgzmjdcnjmwKtzdCMC1kRg5yUAbvzW2R3sZrq6NUi/iZuLZkLxuq2F/36boMovGY
IOOr5MGrwmU48L2sHqOMquS6dVSeDnl6tKiDwGo6qM3aXYVTX1l1zoY=
----END RSA PRIVATE KEY----
```

## 2. Install CLOUDFLOW so that it uses SSL

You can set up CLOUDFLOW to use SSL by specifying the --ssl option with the .pem file obtained in step 1.

Examples:

nucleusd --install --ssl cert.pem

This makes CLOUDFLOW listen to port 9090 using SSL with the default options. You will also need to specify the Webserver URL in the settings page to https://server\_address:9090/

nucleusd --install -i serverid -d mongo ip -p 443 --ssl cert.pem

This makes CLOUDFLOW listen to port 443 using SSL with a custom server id and mongoDB IP address. You will also need to specify the Web server URL in the settings page to https://server\_address:443/

Note: When SSL has been set up, CLOUDFLOW will only accept SSL connections.

# Multiple CLOUDFLOW server setup with fail-over

With a multiple CLOUDFLOW server installation you can balance heavy loads of CLOUDFLOW processes and resources and increase the processing capacity.

It is recommended to provide a fail-over solution, which prevents the prepress workflow from going down in case of a CLOUDFLOW server failure or a MongoDB failure. The recommended fail-over solution in MongoDB is providing **replication sets**.

See CLOUDFLOW configuration on page 25 for more information on how to configure CLOUDFLOW to run with replication sets.

See Example of a setup on page 26 for a fail-over setup example on a Windows system.

## **CLOUDFLOW** configuration

The recommended way to provide fail-over in MongoDB is setting up replication sets.

Replication sets are supported by CLOUDFLOW. During the installation of the CLOUDFLOW server, you can set it to connect to a replication set instead of only to the primary database. In this case CLOUDFLOW will automatically connect to the secondary MongoDB when the primary MongoDB becomes unavailable.

To properly set up CLOUDFLOW to run with replica sets, CLOUDFLOW has to know the IP-addresses of all the nodes in the set (or at least the non-arbiter nodes) and the names of the nodes. You can do this by passing all these values to the **-d** flag when installing the work server.

## Note:

=

- Separate each IP should by a semicolon (;).
- Do not use spaces.
- When you do not specify a port, 27017 is used.
- For Windows:

In clustered CLOUDFLOW setups that share access to **remote file shares via SMB**, you need to configure Windows SMB options on all Windows based cluster members. Set the following keys in the registry in HKLM\System\CurrentControlSet?\Services\LanmanWorkstation? \Parameters\:

- "FileNotFoundCacheLifetime"=dword:00000000
- "DirectoryCacheLifetime"=dword:0000000
- "CacheFileTimeout"=dword:0000000
- "FileInfoCacheLifetime"=dword:0000000

The replication set name is passed by adding the entry replset: \$name\$ to the list of IP's.

#### Example

The replication set is named **cf**. There are three database involved, one running at 10.0.2.1, one running at 10.0.2.2 and one running on 10.0.2.4:27033. This would lead to the following install command:

nucleusd.exe -install -d 10.0.2.1;10.0.2.2;10.0.2.4:27033;replSet:cf

See also the MongoDB documentation for more information on how to set up these replication sets: https://docs.mongodb.com/v3.2/tutorial/deploy-replica-set/.

## Example of a setup

This example explains how to set up a multiple CLOUDFLOW Server installation with fail-over on a Windows system.

## Dual CLOUDFLOW setup with MongoDB Replica set and Arbiter



## Installation of MongoDB server 1 and 2 with replica set

A replica set in MongoDB is a group of mongod processes that maintain the same data set. Replica sets provide redundancy and high availability. They are the basis for all production deployments where fail-over is needed. In case of a fail-over where the Master Database fails, the Replica will become the new Master Database. This means that the replica stays the Master Database from this moment on, until the setup goes in failure again.

How to install this setup?

1. Install MongoDB via the Microsoft Installer Wizard (MSI).

See MongoDB Installation on page 31 for more information.

2. Create mongod.conf file without text formatting.

The content of the file:

```
dbpath = D:\Packz\MongoDB\data
logpath = D:\Packz\MongoDB\log\mongo.log
```

- 3. Run Windows Command Prompt as Administrator.
- 4. Go to MongoDB\bin\ folder.
- 5. Execute the following command line:

```
mongod.exe --install --config D:\Packz\MongoDB\mongod.conf --replSet cf/
ip.address.other.mongodb
```

#### Example

```
mongod.exe --install --config D:\Packz\MongoDB\mongod.conf --replSet
cf/192.168.0.20:27017
```

- 6. Perform step 1 and 5 on both MongoDB servers. Make sure CF/ip.address.other.mongodb point to each other.
- 7. Go to MongoDB\bin\ folder on one of the MongoDB servers.
- 8. Execute Mongo shell by executing mongo.exe on Command Prompt.
- 9. On the Mongo shell cmd line, execute the following command line:

> rs.initiate()

10. On the Mongo shell cmd line, execute the following command line:

> rs.status()

This will tell you which MongoDB server is Primary or Secondary. This is important to set up the MongoDB Arbiter.

## Installation of mongoDB Arbiter on CLOUDFLOW 1 or CLOUDFLOW 2

In case of a fail-over, the Arbiter is necessary to determine whether the MongoDB is unreachable because of a network issue or because of a server shut down. In the event of network issues and shut down, the Arbiter can take the decision to make one of the MongoDB servers the Primary one.

How to install this setup?

1. Install MongoDB via the Microsoft Installer Wizard (MSI).

See MongoDB Installation on page 31 for more information.

2. Create mongod.conf file without text formatting.

The content of the file:

dbpath = D:\Packz\MongoDB\data
logpath = D:\Packz\MongoDB\log\mongo.log

- 3. Run Windows Command Prompt as Administrator.
- 4. Go to MongoDB\bin\ folder.
- 5. Execute the following command line:

```
mongod.exe --install --config D:\Packz\MongoDB\mongod.conf --replSet cf
```

## Example

mongod.exe --install --config D:\Packz\MongoDB\mongod.conf --replSet cf

- 6. Go to MongoDB\bin\ folder on the Primary MongoDB server. See step 10 of previous installation procedure.
- 7. Execute Mongo shell by executing mongo.exe on Command Prompt.
- 8. On the Mongo shell cmd line, execute the following command line:

> rs.addArb("ip.of.arbiter")

With this line you declare to the Primary MongoDB Server that there is a MongoDB Arbiter.

> rs.status()

You should now see the Primary, Secondary and Arbiter setup information.

## Installation of CLOUDFLOW server 1

- 1. Make sure that there is no other instance of CLOUDFLOW running on the system where you want to install it.
- 2. Login to the download center and download CLOUDFLOW.
- **3.** Extract the software to the correct location. The recommended installation location for CLOUDFLOW is recommended by the system:

C:\Program Files\Cloudflow\

4. Run the Command Prompt as Administrator and execute the following command:

```
cd C:\Program Files\Cloudflow
.\nucleusd.exe --install -d
ip.address.of.primary.mongodb.server;ip.address.of.secondary.mongodb.server;replSet:cf
net start cloudflow
```

#### Example

The replication set is called **cf**. There are two databases involved, one running at 10.0.2.1, one running at 10.0.2.2. The IP of the Arbiter does not need to be added. The command would be:

nucleusd.exe -install -d 10.0.2.1;10.0.2.2;replSet:cf

## Installation of CLOUDFLOW server 2

- 1. Make sure that there is no other instance of CLOUDFLOW running on the system where you want to install it.
- 2. Login to the download center and download CLOUDFLOW.
- **3.** Extract the software to the correct location. The recommended installation location for CLOUDFLOW is recommended by the system:

C:\Program Files\Cloudflow\

4. Run the Command Prompt as Administrator and execute the following command:

```
cd C:\Program Files\Cloudflow
.\nucleusd.exe --install -d
ip.address.of.primary.mongodb.server;ip.address.of.secondary.mongodb.server;replSet:cf
-i PP_WORK_SERVER_2
net start cloudflow
```

## Example

The replication set is named **cf**. There are two databases involved, one running at 10.0.2.1, one running at 10.0.2.2. The IP of the Arbiter does not need to be added. The command would be:

nucleusd.exe -install -d 10.0.2.1;10.0.2.2;replSet:cf -i PP WORK SERVER 2

## **CLOUDFLOW WORK SERVERS**

In CLOUDFLOW, processing is managed via the Work Server and multiple Workers. See the topic on WORKSERVERS for more information.

- 1. Open the Work Servers tab in CLOUDFLOW Server 1 > Settings, you will see PP\_WORK\_SERVER. This Work Server represents the Workers that are available on CLOUDFLOW Server 1.
- 2. If you scroll to the bottom of the page, you will see an additional Work Server: **PP\_WORK\_SERVER\_2** which represents the Workers that are available on CLOUDFLOW Server 2.
  - Initially, the Workers list for PP WORK SERVER 2 is empty.
  - In an ideal fail-over solution, CLOUDFLOW Server 1 and CLOUDFLOW Server 2 should have the same Workers list. To add Workers on the PP\_WORK\_SERVER\_2, follow this procedure:
    - a. Select ADD WORKER next to the PP WORK SERVER
    - **b.** Select a Worker from the drop-down list.
    - c. Select the Save icon.
    - **d.** Enable the Worker by setting the switch to **ON**.

# Install extra Work server

Installing an extra Work server provides additional resources.

A Work server is an independent CLOUDFLOW server that is installed on a hardware server or virtual image. It executes the CLOUDFLOW processing through multiple background processes which are called **Workers**. See the topic on WORKSERVERS for more information.

An additional work server can be configured to provide additional resources to a configuration. The reason to add this additional performance power can be various. For example:

- Additional performance for PROOFSCOPE, PACKZflow, RIP... on a separate server.
- More filestore dedicated Workers.
- More handling of web requests.

How to install an extra Work server:

## 1. Make sure you have the correct licenses

Make sure the primary server has the right license key configured. In the list of licenses, an additional work server needs to be available. If this is not the case, contact your local sales representative.

If the license is correct you will see the name of the additional work server. This name needs to be entered in the command line (see next step).

## 2. Install the additional Work server

Analog with the primary Work server installation, the installation can be started via the command line.

For the second work server, an additional argument needs to be added to the command.

**Tip:** Command **nucleusd** --help lists all the possible arguments.

The command for the installation of an additional work server is the following:

```
nucleusd --install -i [name of the workserver] -d [database ip[:port]]
```

#### Example

```
nucleusd --install -i second work server -d [database ip[:port]]
```

## 3. Configure the additional Work server

Configure the Workers for the Work server in SETTINGS > WORK SERVERS.



**Note:** For Windows: Example of a setup on page 26

## **Backup Best Practices**

Back up methods for CLOUDFLOW server role deployments. Server roles may span separate servers or can be combined on a single server.

## MongoDB Database Server

Backing up MongoDB is important. CLOUDFLOW does not provide a feature to (automatically) back up MongoDB, so you need to make sure to regularly back up MongoDB yourself.

Hybrid advises to back up to an external drive or network location. If there is a fatal hardware failure on the database server or the underlying virtualization host, the snapshot data should be unaffected.

Back ups can be achieved for example by using VMware snapshots or Amazon AWS volume snapshots. It is also possible to set up file system snapshots. Two main backup scenarios can be distinguished. It must be understood that these are highly dependent on your specific IT environments.



**Warning:** It is strongly advised that you back up your data applying at least one of these methods or a more comprehensive alternate secure process of your choice. If no valid back up is available support or restores are impossible.

## 1. MongoDB Back Up with Filesystem Snapshot

This MongoDB deployment backup method makes a copy of MongoDB's underlying data files using snapshots from your OS.

With file system snapshots, the operating system takes a snapshot of the volume where MongoDB stores its data files to use as a baseline for data backup. If this volume supports point-in-time snapshots, they can be used to create backups at an exact moment in time.

The mechanics of snapshots depend on the underlying storage system.

## General requirements

- Automatic operating system backups permitting snapshots
- MongoDB journaling enabled
- MongoDB journal located on same logical volume as MongoDB data files

back ups	🗳 back ups
Use Windows' <b>File History</b> to set Windows to make automatic backups. Proceed along the default method:	Use MAC's <b>Time Machine</b> to set MAC to make automatic backups. Proceed along the default method:
<ul> <li>https://support.microsoft.com/en-us/windows/ backup-and-restore-in-windows-10-352091d2- bb9d-3ea3-ed18-52ef2b88cbef</li> <li>Note: Win10 backups are snapshots. File History allows you to define the backup's type (partial of full), frequency and storage location.</li> </ul>	<ul> <li>https://support.apple.com/en-us/HT201250</li> <li>Note: MAC backups are snapshots. Time Machine allows you to define the backup's type (partial of full), frequency and storage location.</li> </ul>

## Enable MongoDB journaling and Journal folder:

To get a correct snapshot of a running mongod process, you must have journaling enabled and the journal must reside on the same logical volume as the other MongoDB data files. Without journaling enabled, there is no guarantee that the snapshot will be consistent or valid.

- To enable MongoDB journaling, start mongod with the --journal command line option.
- To set these configurations automatically, install CLOUDFLOW with will the --launchmongo command line option.

Navigate to your local MongoDB data files location and check for presence of the Journal folder.

MongoDB data files locations:



## How To Back Up a Standalone MongoDB Deployment:

MongoDB has a utility called mongodump for creating a binary export of the contents of a database. Mongodump can export data from either mongod or mongos instances; i.e. can export data from standalone and server deployments.

It is an efficient tool for backing up small MongoDB deployments, but not ideal for capturing backups of larger systems.

• mongodump is run from the system command line (not the mongo shell).

The simple mongodump command connects to the MongoDB instance on the local system on port 27017 and creates a database backup named dump/ in the current directory.

The --out=<path>, -o=<path> command specifies the directory where mongodump will write BSON files for the dumped databases.

Other export options include commands for archiving, zipping, db selections, password setting, etc.

## How To Back Up a MongoDB Server Deployment:

For server deployments more apt methods are MongoDB Cloud Manager or Ops Manager.

MongoDB Cloud Manager is a hosted back up, monitoring, and automation service for MongoDB and has a graphical user interface. It continually backs up MongoDB replica sets by reading the oplog data from your MongoDB deployment and creating snapshots of your data at set intvals.

Ops Manager allows MongoDB subscribers to install and run the same core software that powers MongoDB Cloud Manager on their own infrastructure.

Ops Manager backs up the data as a ongoing and continuous process. This process continues creating snapshots as long as the head database remains synchronized with the database.

## 2. Manual MongoDBback up by copying MongoDB data files

This MongoDB backup method requires manual intervention by the user to make a copy of MongoDB's underlying data files and store this onto another system.

Backups produced by copying the underlying data do not support point in time recovery.

**Warning:** Since copying multiple files is not an atomic operation, you must stop the MongoDB server before copying the files. Otherwise, you will copy the files in an invalid state.

Note:

More details on how to back up your MongoDB installation can be found on the MongoDB website.

## **CLOUDFLOW Application Servers**

CLOUDFLOW application servers do not store any critical information on the local disk. Servers can be redeployed by reinstalling the CLOUDFLOW application software. Point-in-time snapshots may be used for ease of use.

## **CLOUDFLOW Filestores**

This back-up strategy does not include any data stored on CLOUDFLOW filestores. Consult your infrastructure or backup vendor to obtain a backup strategy that is tailored towards your infrastructure and organization.

# **MongoDB** Installation

CLOUDFLOW uses MongoDB to store its information.

MongoDB comes **bundled** with the standard CLOUDFLOW installation. This is the **recommended** way of using MongoDB.

Only in **special cases**, MongoDB can be installed on a separate machine. This is the **alternative** method. See Installation of standalone MongoDB on page 32 for more information on how to do this.

## Run MongoDB as part of CLOUDFLOW (RECOMMENDED METHOD)

In simple setups, you can run MongoDB on the same machine as the CLOUDFLOW server.

To do this, make sure you add the --launchmongo parameter to nucleusd, either when starting it or when installing it. This method will place the database on the system drive.

See nucleusd on page 21 for more information.

## Install MongoDB on a separate machine (EXPERT METHOD FOR SPECIAL CASES)

Only in special cases, can MongoDB be installed on a separate machine.

Possible special cases:

- When a lot of memory usage is expected from the CLOUDFLOW server. For example in case of large RIP tasks, heavy PACKZflow processing or a lot of PROOFSCOPE pre-rendering.
- IT setup preferences:
  - When MongoDB is installed on a separate machine/instance, the database can be managed as a separate IT resource. Consequently, separated CPU, Memory and HDD space can be allocated to it.
  - When configuring CLOUDFLOW for load-balancing or a high-availability setup it makes sense to install MongoDB on a separate box from the CLOUDFLOW Application server.

## System requirements for standalone MongoDB installations

These are the system requirements for standalone MongoDB installations:

A Windows 2012 Server 64-bit system or a Linux server with:

- 8GB RAM
- Intel Core i7 of Xeon E3-1230 (4 cores, 8 Threads)
- Minimum of 500GB of disk space. However, the required disk space depends on the amount of data that will be indexed into the system. If **PROOFSCOPE** is used, the database needs more disk space to store the pre-rendered tiles.

## Guideline to calculate required disk space for PROOFSCOPE:

Per file and per page, 85 lossless compressed tiles per separation + 85 composite tiles are generated.

For example CMYK PDF of 1 page means 5\*85= 425 tiles, where the typical size of a tile is 50Kb. This means that this file approximately requires 20MB in MongoDB.

## Interaction with firewall and anti-virus

MongoDB uses **TCP port 27017**, and binds to the IP address of the machine. If **firewall** or **anti-virus** software is installed on the same machine, make sure that this software is configured appropriately in order not to interfere with the TCP communications between CLOUDFLOW and MongoDB.

## Installation of standalone MongoDB

To install MongoDB, you need to download it first.

You can download MongoDB from http://www.mongodb.org/downloads (Community Server).

- Recommended version: **3.6**.
  - Note: When upgrading to version 19.06 update 2, you must have at least version Sprint 18.10 Update 2.
- Recommended data model: WiredTiger.
  - **Note:** WiredTiger is default from MongoDB version 3.6. However, if you need to upgrade an existing installation of MongoDB to WiredTiger, see Upgrading MongoDB to WiredTiger on page 35 for more information.

## Installation on Windows

## Installing MongoDB on Windows

To install MongoDB on Windows, follow these steps:

- 1. Locate the downloaded MongoDB .msi file (typically located in the default Downloads folder).
- 2. Double-click the .msi file.
- **3.** A set of screens will appear to guide you through the installation process. If you choose the **Custom** installation option, you can specify an installation directory.



Note: These instructions assume that you have installed MongoDB to C: \mongodb.

MongoDB is self-contained and does not have any other system dependencies. You can run MongoDB from any folder and you can install it in any folder (for example D:\test\mongodb).

## Setting up the environment

To set up the environment in Windows, follow these instructions:

Create the config file (for example at C:\mongodb\mongo.conf) with the following contents:

```
systemLog:
  destination: file
  path: "C:\\mongodb\\log\\mongo.log"
  storage:
  dbPath: "C:\\mongodb\\data\\"
```

♦

**CAUTION:** Make sure to use a text editor that can handle newlines. Examples: **Notepad++**, **SciTE** (if you don't have technical permission to install software on the system),...

- **CAUTION:** This file format is white space sensitive. Make sure to use spaces instead of tabs for indentation.
- **Note:** You are free to choose different folder locations, but make sure they all exist or MongoDB will fail to launch.

## Installing MongoDB as a service

To install MongoDB as a Windows service, follow these instructions:

Run a Command Prompt as Administrator and enter the following command:

C:\mongodb\bin\mongod.exe --config C:\mongodb\mongo.conf --install

## Starting and stopping MongoDB as a service

There are two methods to start and stop MongoDB as a service on Windows.

1. Run a Command Prompt as Administrator and type one of the following commands:

- net start MongoDB
- net stop MongoDB
- 2. Open the Services panel from the Task Manager and start/stop the MongoDB service from there.

## Removing MongoDB

To remove MongoDB, follow these instructions:

Run a Command Prompt as Administrator and enter the following command:

C:\mongodb\bin\mongod.exe --remove

## Installation on MacOS

To install MongoDB on , follow the instructions provided on the MongoDB installation site.

https://docs.mongodb.com/v3.2/installation/

To make sure MongoDB does not exceed software limits on **W**, follow these steps:

- 1. Create two plist files and place them in /Library/LaunchDaemons/:
  - a. /Library/LaunchDaemons/limit.maxfiles.plist
  - **b.** /Library/LaunchDaemons/limit.maxproc.plist
- **2.** Copy the following content in the files:

## For maxfiles:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
       "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
 <dict>
   <key>Label</key>
   <string>limit.maxfiles</string>
   <key>ProgramArguments</key>
   <array>
     <string>launchctl</string>
     <string>limit</string>
     <string>maxfiles</string>
     <string>524288</string>
     <string>524288</string>
   </array>
   <key>RunAtLoad</key>
   <true/>
   <key>ServiceIPC</key>
   <false/>
 </dict>
</plist>
```

#### For maxproc:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple/DTD PLIST 1.0//EN" "http://www.apple.com/
DTDs/PropertyList-1.0.dtd">
 <plist version="1.0">
   <dict>
      <key>Label</key>
       <string>limit.maxproc</string>
      <key>ProgramArguments</key>
       <array>
          <string>launchctl</string>
          <string>limit</string>
         <string>maxproc</string>
         <string>2048</string>
          <string>2048</string>
       </array>
      <key>RunAtLoad</key>
       <true />
      <key>ServiceIPC</key>
       <false />
    </dict>
 </plist>
```

3. Change ownership of the files to root:wheel:

```
a. sudo chown root:wheel /Library/LaunchDaemons/limit.maxfiles.plistb. sudo chown root:wheel /Library/LaunchDaemons/limit.maxproc.plist
```

**4.** Reload the files:

```
a. sudo launchetl load -w /Library/LaunchDaemons/limit.maxfiles.plist
b. sudo launchetl load -w /Library/LaunchDaemons/limit.maxproc.plist
B.bastwurmashing
```

5. Reboot your machine.

## Installation on Linux

To install MongoDB on Linux, follow the instructions provided on the MongoDB installation site.

https://docs.mongodb.com/v3.2/installation/

To make sure MongoDB does not exceed software limits on Linux, you need to change the ulimit settings. ulimit refers to the per-*user* limitations for various resources.

To change ulimit settings, issue a command in the following form:

ulimit -n <value>

## Warning:

According to your system's configuration, any change to system limits made by using ulimit may revert after a system restart.

Check your distribution and operating system documentation on how to make the change persistent.

## Example

To limit the maximum open files to 64000, issue ulimit -n 64000 in the config file.

## Upgrading MongoDB to WiredTiger

To upgrade an existing MongoDB Install to the WiredTiger storage engine, follow these steps:

## 1. Stop CLOUDFLOW

- 1. Stop CLOUDFLOW. Don't restart it until you finished the MongoDB upgrade.
- 2. Open an activity/task monitor and verify that all CLOUDFLOW processes are stopped. If there are **nucleusd**, **nucleusweb**, **quantum\*** or **portal.cgi** processes still active after a minute, stop them.

## 2. Upgrade to the latest MongoDB

- 1. Stop the current MongoDB service.
- 2. Download the latest installer for your platform from mongodb.org.
- 3. Run the installer.
- 4. Choose **Custom Installation** if you want to install the software to a location different from the default program files location.
- 5. Locate where your mongod.conf file is situated.
- 6. Register the new MongoDB as a service:
  - a. Make sure the Services panel is closed on Windows.
  - **b.** Open a command prompt as administrator.
  - c. Navigate to the bin folder of your new MongoDB install.
  - d. mongod.exe --remove
  - e. mongod.exe --install --config PATH TO MONGOD CONF
- 7. Start the MongoDB service.
- 8. Verify you are running the correct version by executing from the Mongo bin folder:
  - a. mongo.exe
  - **b.** From the mongo prompt run db.version()

## 3. Convert the Dataset to the WiredTiger storage model

- 1. From the bin folder of your new mongo install run the following to dump your database: mongodump.exe -- out TEMP PATH FOR MONGO DUMP
- 2. Create a new data directory for the WiredTiger storage

**3.** Change the config file to point to the new data directory and force it to use the WiredTiger storage model. This config file must use the YAML syntax. Typical contents for the mongod.conf file:

```
systemLog:
  destination: file
  path: "C:\\MongoData\\logs\\mongodb.log"
  logAppend: false
  storage:
  journal:
   enabled: true
  dbPath: "C:\\MongoData\\db"
  engine: wiredTiger
```

#### Important:

- Don't copy-paste this, but type it fresh, the YAML format is extremely sensitive for hidden characters.
- Only use spaces, no tabs for indentation.
- Escape your backslashes for Windows paths as in the example above.
- 4. Restart the MongoDB service
- 5. Restore your dump to the new MongoDB database: mongorestore.exe TEMP PATH FOR MONGO DUMP

## 4. Your MongoDB is now running on WiredTiger.

It's safe to start CLOUDFLOW again.

## **Repair MongoDB**

In case MongoDB crashes and you cannot start it again without errors, you need to repair the data files.

To do this, you can use the following repair command:

mongod --dbpath /data/db --repair

**Note:** You can include the **--repairpath** option to specify an alternate temporary directory.

When completed, the dbpath should contain the repaired data files and an empty **mongod.lock** file. If you want your current database to be replaced by the repair, you can use your current dbpath after the —dbpath flag.

## In case repair is not working

- When the database has crashed, some data is stored in its journal. This is recovered on startup of the database. Ideally, shut down CLOUDFLOW, start MongoDB and stop it again cleanly before doing a repair.
- Repair MongoDB doesn't work with **journaling** enabled. In case you defined a config file in the repair command containing the following lines, you need to remove them:

```
journal:
enabled: true
```

After MongoDB is repaired, it will start and launch with Journal in case it is the default behavior. Nevertheless, the default is without journaling.

## Back up MongoDB

How to back up MongoDB in Cloudflow.

Backing up MongoDB is important. CLOUDFLOW does not provide a feature to (automatically) back up MongoDB, so you need to make sure to regularly back up MongoDB individually.

#### Note:

More details on how to back up your MongoDB installation can be found :
- in the Backup Best Practices on page 29 topic of this manual.
- on the MongoDB website.

# **CLOUDFLOW** Configuration

CLOUDFLOW needs configuration before you can start using it.

See also the comprehensive CLOUDFLOW User Manual for further details.

Most configuration is done in the SETTINGS tab. It consists of the following tabs:

# DASHBOARD

This tab provides you with details concerning the user login, the database and the Worker statuses.

You can select each section and Worker to see the details.

# LOGGING

This tab shows information about the logs on the files and assets.

You can filter the logs on **Time**, on **User** and on **Log Level**. If you switch on the **Live Updating** button, the logs will be updated immediately without refreshing the page.

# WORKABLES

This tab provides an overview of the existing workables.

You can sort them on **START TIME**, **STATE**, **NAME**, **WORKFLOW** and **CURRENT NODE**. To do so, select the heading.

You can search for workables by entering a search string in the search window of the NAME, WORKFLOW and CURRENT NODE categories.

If you hover over a workable it becomes blue. If you select it consequently, you are directed to the WORKABLES tab.

# FILE STORES

In CLOUDFLOW, files are located in File stores.

You can set up different **File stores** which you can connect to your existing file servers. Therefore, you can continue working with the file server setup you are used to.

File stores have a unique name which is free to choose and a unique location. They can be located on a **local** or on a **remote** drive.

# File store requirements

- A File store for CLOUDFLOW that is mapped to a network share needs to be on the local LAN.
- Indexing should yield 100 files per second at least, if not, the connection is too slow and other processes will run unstable.
- A filestore setup that uses a mountpoint through a WAN or via a VPN tunnel is not supported.

# Add a File store

In SETTINGS > FILE STORES, you can add a File store.

- **1.** Log in as Administrator.
- 2. Open the SETTINGS tab.
- **3.** Open the **FILE STORES** tab.
- 4. Select ADD FILESTORE.
- 5. Enter an internal name for the new File store. The name can only contain alphanumeric characters and underscores.
- 6. Enter a description for the new File store.
- 7. Select the Save icon.

# File storage

CLOUDFLOW uses File stores to map files stored on a specific location. Files can be stored locally, on a network share or on an external location.

# Local storage

The easiest way of storage is local storage or direct attached storage (DAS). In this case, the files are stored on the server itself, and CLOUDFLOW can access the files via a local path. The File store mapping will be a local path, for example C: Filestore.

# On a network share

You can also store your files on a network share.



- UNC paths: on Windows it is preferable to use UNC paths (\\server\share).
  - **Note:** Do not use mapped drive letters, as a mapped drive letter is bound to a user context. The CLOUDFLOW service will run under another user, and will not have access to these drive letters.
- **Mount credentials**: if the server runs as a domain user, the IT department can setup the authorization for the user in such a way that it can mount the network shares without requiring a password. For environments that do no use a domain, CLOUDFLOW offers the ability to specify the mount credentials that need to be used to mount the share.

- SMB share: to connect to a shared folder, you need to use the smb notation (smb://server/share) and restart CLOUDFLOW to mount.
- AFP share: to connect to a shared folder, you need to use the afp notation (afp://server/share) and restart CLOUDFLOW to mount.

=

Note: In case of a change in a remote File store, CLOUDFLOW needs to be restarted in both cases.

# External files

Although it is preferable to use a CLOUDFLOW URL, files that do not belong to a mapped File store can still be accessed by addressing them directly on the file system. You can do this by an official standard which starts with file:///. This URL can differ depending on the OS CLOUDFLOW is running on.

# SX examples:

- file:///folder\_on\_boot\_disk/file.txt
- file:///disk\_name/folder\_on\_disk/file.txt

file:///volumes/disk name/folder on disk/file.txt



# Windows examples:

- file:///C:/folder on disk/file.txt
- file:///disk name/folder on disk/file.txt

# **OUTPUT DEVICE**

In OUTPUT DEVICE you can define, check and update all pages that are linked to printers.

- Output devices: here you can select an output device from the drop-down list.
- Paper definitions: here you can select a paper definition from the drop-down list.
- Print Test Page: with this button you can print a test page. •
- Add Paper Definition: with this button you can add a new paper definition.
- Id: here you can specify an ID for the paper definition or update an existing one.
- Name: here you can specify a name for the paper definition or update an existing one.
- Media type: here you can select the media type (roll or sheet).
- Width: here you can select the media width.
- Height: here you can select the media height. In case you have selected Roll Media, this parameter is not available.

#### Margins

- Top: here you can specify the top margin.
- **Right**: here you can specify the right margin.
- Bottom: here you can specify the bottom margin.
- Left: here you can specify the left margin.

Note: The margin units correspond with the units you have defined in SETTINGS > SETTINGS > = INTERNATIONAL.

- Add: with this button you can add the settings to the paper definition. This button is only available in case you are creating a new paper definition.
- Edit: with this button you can edit the settings of the paper definition. This button is only available in case you are editing an existing paper definition.
- **Reset**: with this button you can reset the settings of the paper definition in case they have not been saved yet.

# FOLDER MAPPINGS

In FOLDER MAPPINGS you can map file paths in a notation to file paths in another notation and change the root of the notation.

To add a new mapping, follow these steps:

- 1. Select Add.
- 2. Select a notation for the **FROM** mapping. Options:
  - **Posix**: select this option for file paths used on Mac and Linux.
  - UNC: select this option for file paths used on Windows using a forward slash.
  - DOS: select this option for file paths used on Windows using backward slash.
  - URL: select this option for file paths in URL notation (forward slashes, % escape...).
- 3. Select a path for the **FROM** mapping. This path will be mapped to the **TO** path.

- 4. Select a notation for the **TO** mapping. Options:
  - **Posix**: select this option for file paths used on Mac and Linux.
  - UNC: select this option for file paths used on Windows using a forward slash.
  - DOS: select this option for file paths used on Windows using backward slash.
  - URL: select this option for file paths in URL notation (forward slashes, % escape...).
- 5. Select a path for the TO mapping. To this path the FROM path will be mapped to.
- 6. Select Save.

#### Example

You have the following situation:

- A storage that is mounted as /Volumes/CompanyStorage/ on Mac and as \\CompanyServer \CompanyStorage on Windows.
- A file on the root of that volume that is called **test.pdf**, and that uses an external file in the same folder, called **image.psd**.

When you save **test.pdf** in for example PACKZ on Mac, the reference to the image is stored as /Volumes/ CompanyStorage/image.psd. However, if you want to use this reference on windows, it needs to be stored as CompanyServer\CompanyStorage\image.psd. In this case, you have to use the following parameters:

- FROM
  - Notation: Posix
  - Path: /Volumes/CompanyStorage/
- то
  - Notation: DOS
  - Path: \\CompanyServer\CompanyStorage

# **WORK SERVERS**

In CLOUDFLOW, the processing is managed via the Work server and multiple Workers.

A **Work server** is an independent CLOUDFLOW server that is installed on a hardware server or a virtual image. It executes the CLOUDFLOW processing through multiple background processes which are called **Workers**.

To view the workservers, in the left sidebar navigate to Settings > Workers

Selecting a workserver will display its associated Workers.

# Add a Work server

You can add a Work server in SETTINGS > WORKERS.

To add a Work server:

- 1.
  - In the left column click the Add workserver icon  $\oplus$ .
- 2. Enter a name in the field.
- 3. Enable the Work server by setting the switch to Active.
- 4. To specify how this server can be reached by other servers add its Internal URL.
- 5. Click Create.

To modify a Work server:

- **1.** In the left column click the name of the workserver.
- 2. Click the ellipsis and select **Delete** to remove the workserver or **Edit** to modify its **Internal URL** or its **Active** status.
- 3. Click Save.

# **Configure a Work server**

In a Work server, you can add Workers and map File stores.

# Add Workers

Workers are background processes that each have a specific task.

# **Specifications of a Worker**

- Workers process jobs one after the other in parallel, but always within the bounds of the hardware resources.
- Workers are managed by CLOUDFLOW itself: a watchdog process monitors them and intervenes when a Worker is blocked or unresponsive.
- Workers can be visualized in the process monitor of the operating system (Activity Monitor on OSX, Task Manager on Windows). They have a specific process name.

See Workers overview on page 42 for an overview of all the Workers that you can add and configure.

# Add a Worker

To add a Worker, follow these steps:

- 1. Log in as Administrator.
- 2. Open SETTINGS > WORK SERVERS.
- 3. Select ADD WORKER.
- **4.** Select a Worker from the drop-down list.
- 5. Select the Save icon.
- 6. Enable the Worker by setting the switch to ON.

Your newly created Worker is set up to handle specific tasks on the mapped File stores.

**Note:** To view the new Workers in the **DASHBOARD** tab you have to refresh the page.

# Set up a Worker for a specific File store

If you add a Worker to a Work server, the Worker is by default set up for all the accessible File stores. Consequently, the Worker will perform its tasks on the files of all the File stores that are mapped in the Work server. However, to optimize your system, you can set up Workers to handle tasks on a single File store. When a single filestore is specified, the Worker will not perform tasks on the other File stores.

To set up a Worker for a specific File store, follow these steps:

- 1. Select the pencil next to the Worker.
- 2. In the drop-down list, select the File store you want to set up the Worker for.
- 3. Select the Save icon.

## Map File stores

If you map a File store in a Work server, you give the Work server access to the files in that File store. By adding Workers, specific tasks are performed on the files in the File store.

All the File stores you have created in **SETTINGS** > **FILESTORES** are by default listed in the Work server.

To map a File store, follow these steps:

- 1. Log in as Administrator.
- 2. Open SETTINGS > WORK SERVERS.
- 3. Select the pencil next to the File store you want to map.
- 4. Specify the path to the files.



5. If needed, enter a user name and password.

6. Select the Save icon.

The Work server can now correctly map the files in the File store. Consequently, every file in CLOUDFLOW can be uniquely identified by its cloudflow://file name.

#### Set up a File store with or without Workers

If you have set up Workers in the Work server, the tasks of the Workers are by default performed on the files of all the mapped File stores. Setting up specific Workers will make sure that:

• The files in the mapped File stores are indexed. For this, you need to add the FILE INDEXING Worker.

**Note:** Make sure you run only one indexer per File store, running multiple indexers will cause the asset database to contain double assets and other inconsistencies.

- The files in the mapped File stores are **processed**. For this, you need to add the following Workers:
  - the METADATA GENERATOR Worker.
  - the **PREVIEW GENERATOR** Worker.
  - the **PROOFSCOPE RENDERER** Worker.

However, you can also select not to add Workers to the File store. A Work server with mapped File stores but **without** Workers can be used as a file mapping in workflows for saving or retrieving files from (network) locations without indexing the resource.

#### Example

A use case for adding a Work server with a mapped File store and without workers is the **CLOUDFLOW Plug-in Suite**. For each different CLOUDFLOW Plug-in Suite client setup, you need to configure a Work Server with a File store mapping, and without Workers. For example MAC\_CLIENT or WINDOWS\_CLIENT.

See CLOUDFLOW Plug-in Suite on page 419 for more information about the CLOUDFLOW Plug-in Suite.

# Workers overview

You can configure the following Workers in CLOUDFLOW:

## DATAROOM

This worker collects diagnostic information about the CLOUDFLOW system.

Process name	Task
dataroom	To collect diagnostic information.

## **EVENT HANDLER**

Deprecated

#### FILE INDEXING

This Worker indexes a File store.

Process name	Task
indexer	To scan a single File store that has been configured for this Worker. As a result, all files will be registered to the CLOUDFLOW system, and changes on the file share will be monitored.

**Note:** Many commercial SMB servers which are not based on Microsoft Windows do not support file system events. Such servers include Linux servers and NAS systems (for example Synology and QNAP). For those systems, the standard behavior still works.

FILE INDEXING has some options:

- Asset Filter: here you can define an asset filter that will be used as a sub-string match on the full file path of each file that the indexer visits. The indexer will only add files to CLOUDFLOW that match with this filter.
  - **Note:** The asset filter is not a regular expression. It will check if the filter is a sub-string of the complete file path. If this is the case, the file will be added.

#### Examples

- If you specify **.pdf** as filter, only complete file paths that contain **.pdf** in their names will be added. These are typically all the files with extension **pdf**.
- If you have a the following file structure:



If you specify **Final\_Files** as a filter, the indexer will only add the files in the **Final\_Files** subfolders to CLOUDFLOW and skip the files in the **Fonts** and **Resources** directories.

• Regex: if you select this checkbox, you can use a regular expression for defining the asset filter.

**Note:** You can use https://regex101.com as an online regex tester.

- **Delay per File**: here you can define the time that the scanning pauses after every 100 folders, before starting the scanning of the next folder.
- **Optimize for file system events**: if you select this checkbox, the scanning is temporary interrupted in case an asset is added or edited. In this case, the asset is added or edited first, after which the scan continues.

**Note:** This option only works for Windows.

• Minimum pass time: here you can define the time interval between the start of the consecutive scans.

#### GARBAGE COLLECTOR

This Worker will remove a file's assets (PROOFSCOPE tiles, thumb, ...) from the database when the file is removed from disk.

Process name	Task
garbagecollector	To remove the assets (PROOFSCOPE tiles, thumb,) from the database when a file is removed from disk.

#### INTELLIGENT FLEXO

This worker applies smart algorithms that enhance screening for flexo printing.

Process name	Task
intelligent flexo	To apply smart algorithms that enhance screening for flexo printing text and images.

With 1 license 2 Intelligent Flexo workers are allowed.

Without a license no Intelligent Flexo workers are allowed.

The Intelligent Flexo worker is able to run without a RIP worker running

# JAVA WEB APPS HOST

This Worker hosts java web archives (.war files).

Process name	Task
JavaWebAppsHost	To host java web archives (.war files). Maximum one copy of this Worker should run on a single workstation.

#### **Deploying .war files**

To deploy .war files in CLOUDFLOW, follow these steps:

- 1. Create a folder in a File store to put the add-ons. For example cloudflow://PP\_FILE\_STORE/ CloudflowAddons.
- 2. Define the location of this add-on path in SETTINGS > SETTINGS > QUANTUM > ADDONS.
- 3. In this add-on folder, create a folder HostedJavaWebApps. Consequently, you have a folder cloudflow:// PP\_FILE\_STORE/CloudflowAddons/HostedJavaWebApps.
- 4. Put the .war file(s) in this folder. You can do this either using **Windows Explorer** or **Mac Finder**, or you can upload the .war files in CLOUDFLOW through the ASSETS view.
- 5. Configure and activate a JAVA WEB APPS HOST Worker in SETTINGS > WORK SERVERS.
- 6. Use the web service defined in the war file on URL: http://<host name or IP address>:8080/ <your service>. For example, to access the Orchestrator API documentation from the local machine where CLOUDFLOW is running, you need to use http://localhost:8080/media/api.

#### **METADATA GENERATOR**

This Worker collects and calculates metadata for files.

Process name	Task
metadata	To collect and calculate meta information for files. The default behavior of CLOUDFLOW is to automatically collect the metadata for every file that is picked up by an indexer.

# NOTIFICATION WORKER

Deprecated.

#### PACKZFLOW

This Worker manages the PACKZflow nodes in the workflow.

Process name	Task
quantumpackz	To launch a child process called <b>PACKZflow</b> . This Worker manages the PACKZflow nodes in the workflow.

# POSTGRES

This process is the PostgreSQL database server.

Process name	Task
postgres	This process is the PostgreSQL database server. Maximum one copy of this Worker should run on a single site.

# **PREVIEW GENERATOR**

This Worker calculates a thumbnail preview for files.

Process name	Task
preview	To calculate a thumbnail preview for files. It supports PDF, various pixel formats like PNG, JPG, etc CLOUDFLOW will calculate a preview for every file that has received metadata.

#### **PROOFSCOPE LIVE RENDERER**

This Worker calculates real-time rendered tiles for PROOFSCOPE.

Process name	Task
proofscope live renderer	To calculate real-time rendering.

PROOFSCOPE Live Renderer is enabled in two locations:

- Settings > Main > Proofscope > Live Rendering: engage this slider to enable Live Rendering as a general setting.
- In the left **Sidebar**: (2) > **Preferences** > **Live Rendering** > **Enabled** > **Save** to enable **Live Rendering** as a user preference setting.

**E** Note: User vs. General

User settings override General settings.

PROOFSCOPE Live Renderer mode (LRM) is only activated when a certain usage within Proofscope so requires.

Usage

- Deep Zoom: Zooming up to 800% is processed by the Proofscope Renderer. From 801% upwards the Proofscope Live Renderer takes over.
- Layers: PROOFSCOPE Live Renderer supports different layer combinations. For each combination the file is read from disk, and the tiles are stored in the DB and cached in the browser.

## **PROOFSCOPE RENDERER**

This Worker calculates pre-rendered tiles for PROOFSCOPE.

Process name	Task
renderer	To calculate pre-rendered tiles for PROOFSCOPE and to store them in the database.

# QUANTUM COMBINED

This Worker executes base parts of workflows.

Process name	Task
quantumcombined	To execute base parts of workflows.

## QUANTUM DATA

This Worker executes some data specific parts of the workflows.

Process name	Task				
quantumdata	To execute some data specific parts of the workflows.				

#### **QUANTUM JAVA**

This Worker executes some data specific parts of the workflows that use Java technology.

Process name	Task				
quantumjava	To execute some data specific parts of the workflows that use Java technology.				

#### RIP

This Worker is the RIP component.

Process name	Task				
quantumrip	This is the RIP component.				

#### SHARE

This Worker executes SHARE file synchronization.

Process name	Task				
quantumshare	To execute SHARE file synchronization. One or more copies of this Worker can run on a single site.				

#### SHARE SCHEDULER

This Worker schedules SHARE file synchronization.

Process name	Task				
quantumshare	To schedule SHARE file synchronization. One or more copies of this Worker can run on a single site.				

# SOFTWARE UPDATE

This Worker checks for software updates.

Process name	Task					
software update	To check for updates of the software. Maximum one copy of this Worker should run on a single workstation.					

# TECTONICS

This worker processes viewing, inspecting, cropping and stitching one-bit TIFF or LEN files.

Process name	Task
tectonics	To process viewing, inspecting, cropping and stitching one-bit TIFF or LEN files.

# VDP ENGINE

This worker executes Variable Data Printing.

Process name	Task				
VDP Engine	To execute Variable Data Printing.				

# **Default Worker setup**

The **default** Worker setup of CLOUDFLOW is the following:

- 1 FILE INDEXING Worker.
- 1 METADATA GENERATOR Worker.
- 2 PREVIEW GENERATOR Workers.

- 1 GARBAGE COLLECTOR Worker.
- 2 QUANTUM COMBINED Workers.
- 1 QUANTUM DATA Worker.

If **PROOFSCOPE** is enabled:

• 4 **PROOFSCOPE RENDERER** Workers.

If **PACKZFLOW** is enabled:

• 1 PACKZFLOW Worker.

If **RIP** is enabled:

• 1 **RIP** Worker.

# Performance Tuning

In order to tune the performance of CLOUDFLOW and depending on he requirements, it can be interesting to modify the default Worker setup.

# More metadata Workers

If many files need to be processed, it can be beneficial to add more METADATA GENERATOR Workers.

## **More preview Workers**

If many files need to be processed, it can be beneficial to add more **PREVIEW GENERATOR** Workers.

## Workflow tuning

When more processing is needed. it can be beneficial to add more QUANTUM COMBINED Workers .

# SETTINGS

In this tab you can define the general CLOUDFLOW settings.

# GENERAL

• Webserver URL: here you can specify the web server URL.

## INTERNATIONAL

- Language: here you can define the user interface language.
- Length: here you can define the length unit. In the decimal precision field you can define the number of digits after the decimal mark.
- Small Length: here you can define the length unit of small distances. In the decimal precision field you can define the number of digits after the decimal mark.
- **Distortion**: here you can define the distortion unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
- Scaling: here you can specify the (fixed) scaling to use while printing the file.
- Text: here you can define the text unit. In the decimal precision field you can define the number of digits after the decimal mark.
- **Resolution**: here you can define the resolution unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
- **Ruling**: here you can define the ruling unit. In the **decimal precision** field you can define the number of digits after the decimal mark.

# METADATA

- **Process Pages (Of Multi Page Documents)**: here you can define the preview processing for multipage files. Options:
  - Process all pages: if you select this option, all the pages of a multipage document will be processed.
  - **Process pages if number of pages**: if you select this option, the pages will be processed in case the amount is smaller or equal to a predefined number.
- **Resolve References**: Enable this slider to activate reference resolving during metadata extraction. Disable this slider to deactivate reference resolving during metadata extraction. This can increase the metadata processing in cases where there is a firewall blocking SMB access.

# MAINTENANCE

- Clean Up Old Renderings
  - When the database gets larger than: if you select this checkbox, old tile renderings will be removed when the database exceeds a predefined number of gigabyte.
  - When the file was not viewed since: if you select this checkbox, old tile renderings will be removed when the file was not viewed since a predefined number of days.
- Clean Up The Trash Folder
  - **Remove files older than... days**: if you select the checkbox next, all files in the trash folder that are older than a predefined number of days will be removed.
- Clean Up Old Data
  - **Remove finished approvals older than**: if you select this checkbox, approvals will be removed when the approval was finished a predefined number of months.
  - **Remove finished jackets older than**: if you select this checkbox, all jackets that are finished and are older than a predefined number of months will be removed.
  - **Remove notes older than** if you select this checkbox, all notes that are older than a predefined number of months will be removed.

# LOGGING

- Log Level: here you can define the minimum log level that will be shown in case of workflow issues.
- Web Trace: with this option you can debug API calls. Options:
  - **Disabled**: if you select this option, web calls will not be logged.
  - Log slow web calls: if you select this option, web calls will be logged in case they take more than 100 ms.
  - Log all web calls: if you select this option, all web calls will be logged, including crashes.

## MENU

- Use Custom Menu: enable this slider to activate a menu of your choice.
- Menu Location: define the location of the menu of your choice.

# **SMTP SERVER**

- Server Address: here you can define the address of the SMTP server. Enable the checkbox next to SSL if SLL is used.
- Username / Password: here you can define the username and password to log in on the server. Select Set credentials to open the dialog, enter the USERNAME and PASSWORD and select Save.
- Sender e-mail: here you can define the email address of the sender.
- Send test-mail to: here you can define a recipient email address to send a test email.

# THIS SITE

- **Name**: here you can define the name of the site. This information is used when setting up CLOUDFLOW SHARE to identify the site more easily.
- **Description**: here you can define the description of the site. This identification is used when setting up **CLOUDFLOW SHARE** to identify the site more easily.
- URL: here you can specify the URL of the site. This information will indicate how the server can be reached from different nodes in the CLOUDFLOW SHARE setup.

# PROOFSCOPE

- Auto Render On Idle: Enable this slider to render images that have not been rendered yet when the PROOFSCOPE RENDERER Worker is idle.
- Live Rendering: enable this slider to render in real-time.

• Renderer: select either the CLOUDFLOW renderer of the PACKZ renderer.

- **Page Box To Render**: here you can select the page box to render. All items outside of this pagebox will not be rendered.
  - **Note:** If you change this setting to render another pagebox, it applies to the files that are rendered after changing the pagebox to render. All files that were already rendered, will not change.
- JPEG quality: here you can define the visual quality of the file in PROOFSCOPE.
- Anti-alias Rendering: Enable this slider to activate *anti-aliasing* rendering.
- **Download Low-res File Button**: with this option you can define the possibility to download low resolution files in PROOFSCOPE. Options:
  - None: if you select this option, downloading the low resolution file will not be possible.
  - **Downsampled**: if you select this option, only the image objects in the file will be downsampled to a predefined quality when downloaded. The linework objects will remain vectorized.
  - **Rasterized**: if you select this option, both the image and linework objects will be rasterized to a predefined quality when downloaded.
- **Download Notes Report Button**: Enable this slider to download the notes report in PROOFSCOPE.
- Exclude Logs From Notes Report: Enable this slider to exclude the logs for the notes report.
- Text Layer Support: Enable this slider to use the Note-From-Text Tool in PROOFSCOPE.
- Show notes history filter: Enable this slider to display the Notes History Filter in PROOFSCOPE.
- Show notes history: Enable this slider to display the Notes History in PROOFSCOPE.

## RIP

- **Presets Protection**: with this drop-down list you can decide whether an unlock is required or not for other users than the administrator. Options:
  - Free, no Restrictions: if you select this option, unlocking will not be required.
  - Unlock required: if you select this option, all users except the administrator will be required to unlock the settings in the **Presets** and **Curves** tabs in the standalone RIP before they can be edited. The tabs can be unlocked by clicking the lock icon next to the title.
- Clean up RIP JOBS folder: Enable this slider to set the number of days after which the Rip Jobs folder will be cleared.

## ASSET VIEWER

- Assset Limit: here you can set a limit to the number of the assets that are viewed in the ASSETS tab.
- Encode URL
  - Encode the url in the info panel: if you select this checkbox, the URL in ASSETS > INFO is encoded. This means that characters outside the ASCII set are replaced with a % followed by two hexadecimal digits. If you leave this checkbox unselected, the URL is not encoded and will not replace non-ASCII characters.

- **Pagebuilder**: if you select the checkbox **Edit in Pagebuilder**, the option **Create Pagebuilder file** is available in the **Actions** drop-down list in **ASSETS**, which allows to create HTML pages in PAGEBUILDER.
- Selecting: if you select the checkbox Enable selecting in viewer, you select a folder or a file in ASSETS in stead of opening it directly. To open a file or folder in this case, you have to click again. If you leave the checkbox unselected, the file or folder will open directly if you select it.

# JOBS

- Details Page By Job Type For Kiosk: here you can define mappings between Job types and URLs. Depending on the Job type, you can define the URL that is linked to when selecting the link in KIOSK > Jobs > Details link.
  - Type: the Job type.
  - URL: the URL.
    - **Note:** Do not use ? in the URL, since the URL will be cut off at that position.
  - Select **H**/**=** to add/delete a mapping.

# Example

- For prepress Jobs, the user needs to go to cloudflow://PP\_FILE\_STORE/resources/ prepressjobshtmlfile/jobs.html when selecting the link in KIOSK.
- For production Jobs, the user needs to go to cloudflow://PP\_FILE\_STORE/resources/ productionjobshtmlfile/jobs.html when selecting the link in KIOSK.

#### In this case, this would be the configuration:

Mapping 1

- Type: prepress
- URL:cloudflow://PP FILE STORE/resources/prepressjobshtmlfile/jobs.html

Mapping 2

- Type: production
- URL:cloudflow://PP FILE STORE/resources/productionjobshtmlfile/jobs.html

# WORKFLOW EDITOR

- Addons: when an addon is installed, you need to specify the CLOUDFLOW path to the addon. The same goes for JAVA modules that need to be processed by the Java WebApp Host.
- Systems Flows
  - View system workflows and activity: enable this slider to make system workflows visible in the Workflow Editor. System workflows are installed by CLOUDFLOW itself an run behind the scenes. By default they are not visible for users. In case you make them visible, they are displayed in an italic font.
    - Note: Some system workflows have a checkbox Auto update. If you select the checkbox, the workflow will be reset to the default system setting when CLOUDFLOW is updated. This means that all changes to the system workflow are lost with every CLOUDFLOW update. If you want to keep the changes, make sure to leave the checkbox Auto update unselected.
- Editor Protection: here you can manage the protection of the Workflow editor. Options:
  - Free, no Restrictions: if you select this option, the Workflow Editor is unlocked by default.
  - Unlock required: if you select this option, the Workflow Editor is locked by default you need to unlock it before you can make changes.

# PACKZFLOW

- Ink Books:
  - Load shared ink books: enable this slider to automatically load the shared PACKZFLOW ink books. If you leave this checkbox unselected, the shared PACKZFLOW ink books will not be loaded automatically.
- Maximum cores used per PACKZFLOW worker: here you can specify the number of cores each PACKZFLOW uses.

# MARS

- Show MARS UI
  - Enabled: if you select this checkbox, MARS is enabled in the UI.
  - **Note:** This checkbox is only available if you don't have a corporate MARS license.
- SERVER URL: here you can configure the IP address or the host name of the MARS server.
  - **Note:** This option is only available if you have a corporate MARS license.

# LOGIN

**Session timeout**: in this field you can define the number of hours you wish your CLOUDFLOW session to run uninterrupted. Minimum value: 0.1 hrs, maximum value: 876,600 hours. (Values higher than 876,600 hours will be reduced to 876,600 hours. Some browsers may not accept high values and return to the login page.)

# ACTIVE DIRECTORY LOGIN SUPPORT

With these settings you can set up an Microsoft Active Directory authentication to allow users to authenticate against Microsoft Active Directory.

Active Directory logins look like an email address (for example: user@domain.local).

When Active Directory Login is set up, the first time an Active Directory user logs in into the system, a new user is created. In the users page, these logins are marked as **Active Directory** users.

An Active Directory user cannot change his/her password in CLOUDFLOW, this needs to be done on the active directory domain.

- **Note:** The Active Directory user needs permission to obtain its own information from the Active Directory server.
- Server: here you can specify the name or IP address of the Active Directory server.
- Active Directory Domain: here you can specify the Active Directory domain.
  - If the domain part of the logins is the same for everyone, you can specify the domain (including @) in this field. In this case, the user can leave the domain field empty when logging in into CLOUDFLOW, and only Active Directory logins in the specified active directory domain are allowed.
  - If several domains are used, you can leave this field empty. In this case, the user needs to enter the correct domain when logging in into CLOUDFLOW.

**Distinguished Names**: here you can enter the LDAP distinguished names of the users to authenticate. CLOUDFLOW uses these to look for the users in Active Directory.

LDAP distinguished names look like this: cn=User, ou=my-org, dc=my-domain, dc=com

**Important:** Active Directory and LDAP distinguished names are an extensive and complex matter for which you need to have specific knowledge. Always ask your system administrator for help and advice.

In case your system administrator is not able to help you, you can follow these steps to get the LDAP distinguished name for an Active Directory user on Windows:

- 1. Select Start > Run on a Windows system (server or client).
- 2. Enter compmgmt.msc
- 3. Select OK. The Computer Management Console opens.
- 4. Expand Users & Groups.
- 5. Select Groups.
- 6. Open the **Properties** for one of the groups and select Add.
- 7. Select the Users window and select Advanced.
- **8.** In the **Select Users** window, search for the admin user name and select to show the X500 name in the attributes to display. This is the full distinguished name.
- Important: If you enter the distinguished name like in the above example (cn=User, ou=myorg, dc=my-domain, dc=com), it will only return the first user. If you want to add all users, you might need to leave out the first cn component. Based on the example above, this would be ou=myorg, dc=my-domain, dc=com. However, this depends on your AD setup. Ask your system administrator for help.

Several paths can be separated by a new line. Distinguished names will create a *shadow user* in CLOUDFLOW referring to the user in Active Directory.

• **Default Scope**: here you can define the scope to assign to Active Directory users when they log in for the first time.

#### **GOOGLE LOGIN SUPPORT**

With these settings you can activate a Google login in case you want to allow user authentication via Google accounts. To do this, you need to obtain a Google Client ID and Client Secret. See the Google developers website for more information on how to do this.

- Client ID: here you can enter your Google client ID.
- Client secret: here you can enter your Google Client Secret.

#### **UI SETTINGS**

- Home Page: here you can define a custom home page URL.
  - Note: You can also create a home page using a scope, see Create a scope on page 286 for more information. If a scope welcome page is defined, this has priority over the HOME PAGE setting in SETTINGS.
- NEBULA UI: enable this slider to apply CLOUDFLOW's Nebula user interface.

# Inspect Collections

This tab gives an overview of all the database collections.

**Inspect Collections** is read only. To edit the collections, select **Collections** to activate the Edit Mode.

# Parameters

- Inspect Collections: here you can enter a database query. The query needs:
  - To be an array of strings.
  - The same structure as an API call.
  - To be between double quotes.

# Example

["cloudflow.file", "contains", ".pdf"]

For the syntax, see topic General Topic list - query in the API reference (on http://<server>:<port>/?api).

• CLEAN UP ALL STALE APPROVAL(S)<sup>2</sup>: if you select this option, you can clean up all stale approvals. These are approvals that have become useless.

# Example

An approval flow was started but the files are not yet approved or rejected. A user removes the workables of the flow. This leaves assets to be approved, but forcing the approval is not possible because the workables don't exist anymore.

- LIMIT: here you can enter the maximum number of records that you want to display on the screen.
- COLLECTIONS: here you can select the collection you want to open from a drop-down list.

# **Extra functionalities**

- Sort the records by clicking on the headings (= the keys) on top of the records.
- Filter the records by entering a value in the filter fields.
- Delete each record individually by selecting on the right.
- Delete them all at once by selecting **Delete all** at the bottom.
- Select ▶ to expand or collapse the record tree. If you select ૠ and ▶ simultaneously, all record trees will be expanded.
- Double click on a record to view and/or edit the JSON code. If you edit the JSON file, select Save to save it.

# **CLOUDFLOW Plug-in Suite SETTINGS**

In the CLOUDFLOW Plug-in Suite SETTINGS you can define the CLOUDFLOW Plug-in Suite settings.

The **CLOUDFLOW Plug-in Suite SETTINGS** are available if you have installed CLOUDFLOW Plug-in Suite. See **CLOUDFLOW Plug-in Suite** on page 419 for more information.

- **Cloudflow Server URL**: the URL of the CLOUDFLOW Server.
- Direct File Access: if you select this checkbox, the file will be accessed directly from within the File store when you open in via File > CLOUDFLOW > Open From Cloudflow. If you leave this checkbox unselected, the file you open with File > CLOUDFLOW > Open From Cloudflow is copied in the Work Folder and will be accessed from here.

<sup>&</sup>lt;sup>2</sup> This option is only available when inspecting the approval collection

• Client Name: the name of the CLOUDFLOW client, for File store mapping or mounting (for example MAC\_CLIENT or WINDOWS\_CLIENT). This is to make sure that the plug-ins have access to the correct files on the correct File store.

# Example

You have configured two Work servers in CLOUDFLOW:

- **PP\_WORK\_SERVER**, which contains the CLOUDFLOW Work server configuration and a mapping to a local File store (**PP\_FILE\_STORE**).
- MAC\_CLIENT, which also contains a file store mapping PP\_FILE\_STORE to the File store on PP\_WORK\_SERVER.

If you enter MAC\_CLIENT in Illustrator > Window > CLOUDFLOW > Preferences > Client Name, the Illustrator plug-in has access to the files on the File store on PP\_WORK\_SERVER

- User Name: the User name of the CLOUDFLOW user.
- **Password**: the password to log in with the CLOUDFLOW users.
- Enable Automatic Login: if you select this checkbox, you will be automatically logged in.
- Packz Application: the location of PACKZ on the work station.
- Work Folder: the location where the files are temporary stored when they are in a workflow that is put on hold via a Hold in Kiosk node. The files are downloaded to this work folder and are opened from this folder and you can view and edit them. After the file has been edited and released, the file is removed from the folder.
  - **Important:** In case the checkbox Use File Store Access is selected in the Hold in Kiosk node, the file is not stored in this folder. In that case, the file is opened directly from the Upload Location in the File store that you configured in the Start From Kiosk node.

See Open a file that is on hold and release the workable on page 424 for more information about workables that are put on hold.

- Save: select this button to save the settings.
- Go To Login: select this button to go to the CLOUDFLOW log in page.
- Check Cloudflow Server: select this button to sync the file store info and check the status of the CLOUDFLOW server.
- File Store: indicates which File store is known.
- **Path**: indicates the File store path.
- Accessible: indicates whether the File store is accessible or not.

# RESOURCES

In this tab you can define the names of the folders where CLOUDFLOW stores its resources.

CLOUDFLOW suggests default folder names. You can change them if needed.

# PATCHPLANNER

In this tab you can define the settings for the PATCHPLANNER option.

- JOB MARKS PATH: here you can define the path for the job marks. Job marks are also referred to as cylinder marks in PATCHPLANNER.
- FONT PATH: here you can define the path for the fonts used.
- MARKS PATH: here you can define the path for the patch marks.
- MOM OUTPUT PATH: here you can define the output path for MOM XML output file.
- MIRROR PROOFS OUTPUT PATH: here you can define the output path for the mirror proof PDF.
- DRILL MOUNT OUTPUT PATH: here you can specify the output path for the drill mount.
- JOBS PATH: here you can define the path for the jobs.

- PATCH SHEETS PATH: here you can define the path for the sheets.
- **PDF OUTPUT PATH**: here you can define the path for the black color PDF file to be RIPped.
- **PDF OVERVIEW OUTPUT PATH**: here you can define the path for the PDFOverview file.
- FORMAT OUTPUT FOR
  - Default mount method: here you can define the default mount method. Options:
    - Automatic (MOM): select this option in case the patches will be mounted automatically on the carrier in the mounting machine. A MOM XML file that contains the coordinates of the patches will be exported .
    - **Mirror**: select this option in case the mounting of the patches will be done by mirror mount. A mirror proof PDF will be exported. This file is printed and taped onto the proofing drum of the mounting machine. It will display through a mirror and the marks allow for the patched job to be positioned exactly.
    - Drill Mount: select this option in case the mounting of the patches will be done by drill mount.
    - Mirror and Automatic (MOM): select this option if you want to combine the two options above.
  - Cutting method: here you can select the default cutting method.
  - Cut gutter: here you can select the cut gutter.
  - Start layout from: here you can define where the layout on the sheet should be started.
  - Output patch sizes: if you select this checkbox, the patch sizes will be added to the MOM XML file.
- ORIGIN
  - Origin: here you can define the origin of the patches.
  - Orientation: here you can define if the coordinates are right-handed or left-handed.
  - Offset x: here you can define the offset relating towards the x axis.
  - Offset y: here you can define the offset relating towards the y axis.
- SHEET
  - Margins: here you can define the left, right, top and bottom sheet margins.
  - Default Size: here you can define the default width and height of the sheet.

# MAINTENANCE

In this tab you can perform some actions on the database.

# LICENSE

In this tab you can see which distributed licenses are used by which site and you can clear or update your license file.

# HOME

The HOME page contains an overview of all the assets that needs to be approved.

Files with an approval state are indicated by one of the following icons: V for an approved file, X for a rejected file, ? for a pending file and M for a file with multiple approval statuses.

# Side bar

The Nebula Side bar is the gateway to CLOUDFLOW's modules and functionalities.

Via the Nebula **Side bar** the user can access CLOUDFLOW's modules and functionalities. The **Side bar** is customizable.



# **DEFAULT MODULES**

- 1. Home: This item links to the Home page page.
- 2. Notifications: This item links to the Notifications page.
- 3. Assets: This item links to the Assets page.
- 4. Approval: This Item links to the Approval page.
- 5. Tasks: This item links to the Tasks page.
- 6. Jobs: This item links to the Jobs page.
- 7. Rip: This item links to the Rip page.
- 8. Kiosk: This item links to the Kiosk page.
- 9. Flows: This item links to the Workflows page.
- 10. VDP: This item links to the VDP page.
- 11. Tectonics: This item links to the Tectonics page.
- 12. Mars: This item links to the Mars page.
- 13. Dashboard: This item links to the Dashboard page.
- 14. Settings: This item links to the Settings page.
- **15. Manual**: This item links to the Manual page.

# USER BUTTON

The user button reports either carries your name or the role that you hold (e.g. Administrator), and the version of CLOUDFLOW you are running. It also leads to 3 options:

- 1. Log Out: Here you can safely log out of CLOUDFLOW.
- 2. All Scopes: Here you can select a scope that is available for your user profile.

- 3. Preferences: Here you can define user preferences. These override the global settings of #14 above.
  - Localization: Language: here you can define the user interface language.
  - Units:
    - Length: here you can define the length unit. In the decimal precision field you can define the number of digits after the decimal mark.
    - Small Length: here you can define the length unit of small distances. In the decimal precision field you can define the number of digits after the decimal mark.
    - **Distortion**: here you can define the distortion unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - Scaling: here you can specify the (fixed) scaling to use while printing the file.
    - **Text**: here you can define the text unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - **Resolution**: here you can define the resolution unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - **Ruling**: here you can define the ruling unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
  - Change password: Here you can define and change your password.
  - **Proofscope**: Live Rendering: enable this slider to render in real-time.
  - Versions
    - CLOUDFLOW: here you can see your CLOUDFLOW version.
    - **PACKZFLOW**: here you can see your PACKZFLOW version.

## ADD / DELETE MENU ITEMS

In the sidebar links to functionalities can be added and removed.

To do so, follow these instructions:

- 1. On your Windows keyboard, press ALT+SHIFT.
- 2. Move your cursor to the bottom left corner of the sidebar menu.

The User name indication will disappear and a pencil icon will be displayed.



3. Click the pencil icon.

The Edit the menu window is displayed.

	Edit the menu					0 D
$\square$	application	Available	Menu	Icon Type	Icon	~
$\square$	Home			lcon	0.*	
Û	Notifications	Miyakoshi	application Place Queues Cutoffs Statisti Archive Settings	Type	Container	~
	Assets			Custom ID		
0	Approval			Custom ID		
6	Tasks		Notificati.	Label	application	0
≣	Jobs			Tooltip		0
÷.	RIP			User	No Permissions	~
믑	Kiosk 🛛 🕄		Approval EmailAll ApMy ApMy Tea	User		
₽	Flows			Attribute		
៍ំំំ	Intelligent Flexo T		Tasks All Tas My Tas All Cal.			
	Tectonics					
⇒	PatchPlanner		Jobs Config Forms Teams Resour			
Δ	Mars					
G	Dashboard		RIP Jobs Presets Curves Color			
٢	Settings					
0	Manual		ROSK			
		Miya	y Dani			
			Rows			
			Inteligen			
			E Tectonics ·			
2	Administrator		Cancel Save			

- 4. To add an available functionality to the menu, drag it to the middle Menu to a location where you want it to sit in the sidebar menu.
- 5. To delete a functionality from the menu, click its *delete dash* and it will snap back to the Available column.
- 6. Click Save.

# **CREATE / EDIT MENU ITEMS**

In the sidebar the new links to functionalities can be created and existing links can be edited. Note that Default links cannot be edited.

To do so, follow these instructions:

- **1.** Follow the above steps 1 to 3.
- 2. To edit items, select one in the Menu column and proceed with step 4.
- 3.

To **create** items, click the **Add new** button

In the Menu column an item named Application is added. Select it.

- 4. In the pane below select as Icon type either an Icon or an Image of your choice.
- 5. As Type either point to a Link or define a Container
- 6. To name or rename, in the Label field enter the new name.
- 7. To provide a tooltip, in the **Tooltip** field enter a short description.

## =

Note: Both Label and Tooltip should be translated. Click the corresponding globe icon  $\bigcirc$  or  $\bigcirc$  and enter translations.

- 8. Under User Permissions define one or more permissions limiting which users will be able to see/access that same menu item (across scopes).
- **9.** Under User Attribute define one or more attributes limiting that only users with that attribute will be able to see/ access that same menu item (across scopes).

**Note:** User Attributes first appear in blue. Only after you click **Save** do they become active.

10. Click Save.

## RESET MENU

Prior customisation can be easily undone by clicking the Factory Reset button

The Factory Reset buttons leads to 3 options:

- 1. cancel: cancel reset.
- 2. continue editing: current menu items will be replaced by the default menu and the menu stays customised.
- **3.** quit customisation: current menu items will be reset to the default menu and in the main settings the check box customised menu will be deactivated.

# **User Button**

The user button provides a link to multiple pages.

The user button is located at the bottom of CLOUDFLOW side bar.

$\bigcirc$	Dashboard				
¢	Settings				
?	Manual				
		admin v21.4 build	35204	6	
			$\mathbf{\nabla}$	Ğ	
9	Administrator	Logout	All Scopes	Preferences	

The user button either carries your name or the role that you hold (e.g. Administrator), and the version of CLOUDFLOW you are running. It also leads to 3 tabs:

- 1. Log Out: Here you can safely log out of CLOUDFLOW.
- 2. All Scopes: Here you can select a scope that is available for your user profile. See Scopes.
- 3. Preferences: Here you can define user preferences. These override the global Settings.
  - Localization: Language: here you can define the user interface language.
  - Units:
    - Length: here you can define the length unit. In the decimal precision field you can define the number of digits after the decimal mark.
    - Small Length: here you can define the length unit of small distances. In the decimal precision field you can define the number of digits after the decimal mark.
    - **Distortion**: here you can define the distortion unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - Scaling: here you can specify the (fixed) scaling to use while printing the file.
    - **Text**: here you can define the text unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - **Resolution**: here you can define the resolution unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
    - **Ruling**: here you can define the ruling unit. In the **decimal precision** field you can define the number of digits after the decimal mark.
  - Change password: Here you can define and change your password.
  - Proofscope: Live Rendering: enable this slider to render in real-time.
  - Versions
    - CLOUDFLOW: here you can see your CLOUDFLOW version.
    - **PACKZFLOW**: here you can see your PACKZFLOW version.

# **CLOUDFLOW WORKSPACE**

# CLOUDFLOW WORKSPACE is the base module of CLOUDFLOW.

It contains:

- File management.
- A powerful workflow engine for automations.
- User management.
- Scope management.
- Variable processing.
- Approval and collaboration processes.

# Assets

CLOUDFLOW accesses the files on your file servers through **File stores**.

The File stores are indexed. This means that when a file is added to a file server, the software:

- Detects it.
- Creates a thumbnail.
- Extracts the metadata and stores it in the database.

Once a file is indexed, it is an *asset* in CLOUDFLOW. You can browse to it and search it in the ASSETS tab.

# ASSETS tab

The ASSETS tab provides you with an overview of the File stores.

The default File store in CLOUDFLOW is **PP\_FILE\_STORE**. Do not remove this File store.

Select a File store to open it and view the content.

- Note: If the checkbox Enable selecting in viewer in SETTINGS > SETTINGS > ASSET VIEWER > SELECTING is checked, the files and folders in the File store will be selected first and you have to click to open them.
- **Note:** If you're using **SHARE**, you can reach the files and File stores of the different sites by selecting the sites icon in the ASSETS tab (1).

# List View / Icon View

## List View

Select **List View** to view the files and folders of a File store as a list. You can change the column width by using the drag handle which appears when you move the mouse over a column boundary. The following columns are present:

- Name: the name of the file.
- File Type: the file type.
- File Path: the file path.
- Approval: the approval state. All approved, rejected or pending files will have an attribute Approved, Rejected or Pending. If a file has multiple approval states, all according labels are displayed.
- Tags: the file tags. You can add a new or an existing tag to both folder and files by selecting +.
- Last Modified: the modification date and time.

Select the column header to sort the files and folders by column type.

Select Load More to load more files and folders. See ASSET VIEWER on page 49 to edit the displayed asset limit.

To download a file, hover over the file and select 速. The file will be downloaded in high resolution.

#### Icon View

To download a file, hover over the file and select 🕹. The file will be downloaded in high resolution.

Files with an approval state are indicated by one of the following icons: V for an approved file, X for a rejected file, ? for a pending file and M for a file with multiple approval statuses.

When Icon View is displayed, an extra option Sort By is available. With this option you can sort the files by Name, File Type, File Path and Modification Time.

## Upload file...

There are various ways in which you can upload a file to a File store:

- By selecting Upload File...
- By dragging and dropping them into the File store.
- From within PACKZ with the function Save To Cloudflow. See the PACKZ manual for more information.

**Note:** Make sure the **FILE INDEXING** Worker is activated for the File store. If not, the file will not be indexed.

# **Open Folder in OS**

Select Open Folder in OS to open the active folder on the operating system.

**Note:** This option is only available if you have a license for the **CLOUDFLOW Plug-in Suite**. See **CLOUDFLOW Plug-in Suite** on page 419 for more information about this module.

#### Actions

Options:

- Open in Packz: opens the file in Packz.
- Open Folder in OS: opens the folder in the operating system.
- Reveal file in OS: opens the file in the operating system.
- Add folder: adds a folder to the File store.
- Create Pagebuilder file: opens PAGEBUILDER and allows you to create a webpage in HTML format in the selected File store or Files store folder. The HTML webpage created with Pagebuilder is a CLOUDFLOW asset with a CLOUDFLOW URL and can only be used in a CLOUDFLOW environment. It can be for example be used as an upload page, approval page, customized landing page... See Use case of PAGEBUILDER on page 115 for a Page Builder use case.

Note: The option is only available if the checkbox Edit in pagebuilder in SETTINGS > SETTINGS > ASSET VIEWER > PAGEBUILDER is selected.

## Search

=

You can search on all the asset information that is stored in the database, such as metadata, approval information, tags...

# Examples

- The name of the document.
- The file extension.
- The barcode code.
- The MIME type(s) of the document.
- The number of pages.

• ...

#### There are two types of searches:

- **Basic or quick search**: with this search you can enter content in the search field on the right. You can enable or disable automatic search:
  - In case the automatic search icon is selected (\*), the automatic search is enabled. This means that as soon as you enter content in the search field, the files you see will match the entered criteria. When you clear or add content in the search field, filtering will be disabled again.
  - If the automatic search icon is deselected (\*), the automatic search is disabled. This means that after you have entered content in the search field, you have to hit the Enter key first to see the files that match the entered criteria. When you clear or add content in the search field, filtering will be disabled after you hit the Enter key.

# Examples

=

- If the asset you are looking for is 8 pages long, enter 8.
- If you want to look for all assets that have been rejected, enter status:pending.
- If the asset you are looking for contains EAN-13 barcode 4006381333931, enter 4006381333931. If you want to look for all assets that contain EAN-13 barcodes starting with 400638, enter 400638.

**Note:** The search results are not limited to the folder where you are when you start typing, it encloses all the files you have access to.

Advanced search: with this search you can search on all information (colors, file info, ...) and execute sophisticated, combined queries (... AND ... OR ... ).

You can select an existing query from the drop-down list, or create a new query from scratch.

To create a new query from scratch, follow these steps:

- 1. Select **More Options** next to the Search field. It appears when you hover with your mouse on the left side of the field.
- 2. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.
- 3. Select the criteria.
- 4. If you want to add a second rule or group, select AND or OR.
- 5. If needed, define the additional rules or groups.
- 6. Select **Delete** to delete a rule or a group of rules.

## Example

If you want to search for all .pdf files of which the file names contain 8 or 9, you can define the following rule:

1. Rule File Extension equal .pdf

AND

- 2. Group
  - a. Rule File Name contains 8

OR

b. Rule File Name contains 9

## Manage queries

Select 🌣 to open the query management window. To add a new query, follow these steps:

- 1. Select Create new.
- **2.** Enter a name for the query.
- **3.** Add a rule or a group as described above.
- 4. Select Save.

You can now select the newly created query from the drop-down list.

**Note:** You can combine basic and advanced search.

#### **Context menu**

=

You can open the context menu for each folder or file by right clicking. The context menu contains the following options.

In case of a folder:

- Open Folder in OS: opens the folder in the operating system.
- Add folder: adds a folder to the File store.
- Create Pagebuilder file: opens PAGEBUILDER and allows you to create a webpage in HTML format in the selected File store or Files store folder. The HTML webpage created with Pagebuilder is a CLOUDFLOW asset with a CLOUDFLOW URL and can only be used in a CLOUDFLOW environment. It can be for example be used as an upload page, approval page, customized landing page... See Use case of PAGEBUILDER on page 115 for a Page Builder use case.
- Delete: deletes the active folder.
- Cut: cuts the active folder to the clipboard.
- **Copy**: copies the active folder to the clipboard.

**Note:** If there is a folder on the clipboard, open the folder where you want to move/copy it to and select the folder name from the **Actions** menu.

• **Paste**: pastes the folder that is on the clipboard.

In case of a file:

- **Download**: downloads the file.
- **Rename**: renames the file.
- Delete: deletes the file from the File store.
- **Duplicate**: duplicates a file filename.extension in the same folder as filename 1.extension. If the file is a PageBuilder page it duplicates the content according to PageBuilder.
- Cut: cuts the file from its current location to the clipboard to paste it in another location. Open the desired location and select Actions. Select the file you have cut from the menu to paste it.
- **Copy**: copies the file to the clipboard to copy it to another location. Open the desired location and select **Actions**. Select the file from the menu to copy it.
- **Paste**: pastes the file that is on the clipboard.

# Asset details

When you select a file, the asset details are displayed and you can perform some actions on the active file.

You can perform the following actions:

• Replace File...: if you select this option, you can browse to another file on the server and replace the existing one.

**Note:** This option is only available if the user has the permission to upload files. See Manage users on page 283 for more information.

- No Approval / Waiting Approval / Approved / Rejected: if you select this option, more information of the approval state of the asset is shown. See Approvals on page 298 for more information on approvals.
- Reveal File in OS: if you select this option, the folder where the file is located is opened.
  - **Note:** This option is only available if you have a license for the **CLOUDFLOW Plug-in Suite**. See CLOUDFLOW Plug-in Suite on page 419 for more information about this module.
- Open File with Packz: if you select this option, the file is opened in PACKZ.

**Note:** This option is only available if you have a license for the **CLOUDFLOW Plug-in Suite**. See CLOUDFLOW Plug-in Suite on page 419 for more information about this module.

• View File: if you select this option, the file is opened in **PROOFSCOPE**.

- Re-render Asset: if you select this option, the existing asset tiles in the database is re-rendered.
  - **Note:** This option is only available for users with admin rights.
- Actions: with this option you can do the following actions:
  - Download the file.

Note: You can also download a file by hovering over the thumbnail image and selecting 🛃.

- Rename the file.
- **Delete** the file from the File store.
- Cut the file.
- Copy the file.
- Paste a formerly cut or copied file that is on the clipboard.
- **Note:** This option is only available for administrators.
- Compare File: if you select this option, you can compare the file with another file in PROOFSCOPE.
  - **Note:** This option is only available with files that are supported in PROOFSCOPE.

See Supported PROOFSCOPE formats on page 315.

• Edit: if you select this option, you can open Pagebuilder and edit HTML pages that were created in PAGEBUILDER. See Pagebuilder on page 66 for more information. You cannot use Pagebuilder to edit HTML files that were not created in PAGEBUILDER. If you try to edit an HTML file with missing Pagebuilder data, a warning error appears. This option is only available with HTML pages.

The file details contain the following tabs:

# INFO

This tab shows you the following information:

- General information.
- Information about the sizes of the Media box and the Trim box.
- Colorants of the file.
- **Note:** In case of color profiles (\*.icc and \*.icm), the INFO tab contains the type of the profile and the color spaces on both input and output side of the profile.

A special note on Server URL:

All files known by CLOUDFLOW are identified by their specific **CLOUDFLOW path**. This is a URL which uniquely identifies the file. You can use this URL anywhere in the workflow to select the specific file.

Fileserver3 > 01 WIP > files > FD000038 > Olive_Oil_Z34DE.pdf							
	Olive_Oil_Z34DE.pdf Singlepage PDF File Replace File No Approval View File Re-render Asset View Ticket Compare File						
INFO PAGEBOXES	PRODUCTION TAGS APPROVAL WORKFLOW SHARE LOG						
Server URL	cloudflow://Fileserver3/01%20WIP/files/FD000038/Olive_Oil_Z34DE.pdf						
MIME Type	application/pdf						
Pages							
File Size	17 KB						
Creation Time	2016-05-27T09:46:57+0200						
Modification Time	2016-05-09T17:43:43+0200						

- A: the name of the File store where the file is located.
- **B**: the folder path where the file is located.
- C: the file name.

Attention: The space in the path is translated to %20 in the CLOUDFLOW URL.

# PAGEBOXES

This tab shows information about the page boxes.

## PRODUCTION

This tab shows information about the production colorants.

## LAYERS

This tab shows information about the layers.

#### REFERENCES

This tab shows the external references. If the file doesn't contain external references, the tab is hidden.

The column Quality can have the following result:

- Not Resolved: this result is shown when the original reference could not be resolved.
- Full Resolve: this result is shown when the there is not doubt that the resolved URL references the correct item, but it does not mean that the file exists.
- **Partial Resolve**: this result is shown when there is a high degree of certainty that the resolved URL contains the correct item.
- File not found: this result is shown when the reference was resolved when the metadata was calculated, but that the generated CLOUDFLOW URL does not exist when the reference tab is opened.

# XMP

This tab shows information about the XMP data of the file.

# TAGS

In this tab you can add custom tags to the file.

# APPROVAL

In this tab you can perform multiple actions. In case the file has been submitted to an approval workflow, you can perform the following actions:

- You can select an action on the file from the drop-down list:
  - Force Approved: the file will be approved.
  - Force Rejected: the file will be rejected.
  - Cancel: the approval will be canceled.
- You can see the approval state of the file.
- You can see the notes that were added to the file in PROOFSCOPE.

If the file has not been submitted to an approval workflow, you can perform the following actions:

- You can submit the file to an approval workflow:
  - 1. Select the approval flow in Launch Approval > Input.
  - 2. If needed, enter the approval parameters in Approval Details. See Form Builder on page 260 for more information on how to define these parameters.
  - 3. Select Submit.

See Build an approval cycle on page 299 for more information on how to build approval workflows.

- You can invite users or contacts to view the file in PROOFSCOPE:
  - 1. In the section Invite Users To PROOFSCOPE, select the user or contact.
  - 2. If needed, edit the mail message.
  - **3.** Define until when the invitation will be valid.
  - 4. Select INVITE.

## WORKFLOW

In this tab you can submit the file to a workflow:

- 1. Select the workflow in Workflow.
- 2. Select the Input.
- **3.** If needed, enter the workflow parameters. See Form Builder on page 260 for more information on how to define these parameters.
- 4. Select Submit.

## SHARE

This tab shows information about the file share. See CLOUDFLOW SHARE for more information.

# LOG

This tab shows information about the logs of the selected file. You can filter the logs by time, by user and by level. If you switch on the **Live Updating** button, the logs will be updated immediately without refreshing the page.

# Pagebuilder

With Pagebuilder, you can create custom HTML pages.

Pagebuilder is a tool in CLOUDFLOW that allows you to create custom HTML pages. The HTML page is an HTML file that is built up with CLOUDFLOW components.

The HTML page is a CLOUDFLOW asset.

# **Activate Pagebuilder**

You need to activate PAGEBUILDRER before you can use it.

To be able to use PAGEBUILDER, you first need to activate it. To do so, go to SETTINGS > SETTINGS > ASSET VIEWER > PAGEBUILDER and select the checkbox Edit in pagebuilder.

# Create a Pagebuilder page

There are two steps to create a Pagebuilder HTML page.

- **1.** Create the HTML file.
  - a. Go to ASSETS and open the File store and folder where you want to create the HTML page.
  - **b.** Select Actions > Create Pagebuilder file.
  - c. Enter a name in the field next to Save As:.
  - d. Select Save.
- 2. Open the HTML page in Pagebuilder and build up the page with the components. See Build up the page on page 71 for more information on how to use Pagebuilder and Pagebuilder components on page 72 for a complete overview of the components parameters.

# Edit an existing Pagebuilder page

To edit an existing Pagebuilder page, follow these steps:

- 1. Go to ASSETS and select the asset.
- 2. Select Edit.

The page will only open in Pagebuilder if the page is a well-formed HTML file that was built with PAGEBUILDER.

**Note:** If the Pagebuilder page is an asset linked to multiple MARS Apps, a drop-down list appears to select the MARS App.

# Pagebuilder overview

Pagebuilder consists of different areas.



#### A: the components

These are the components you can use to build up the page. See Pagebuilder components on page 72 for a complete overview of the Pagebuilder components.

#### B: the Pagebuilder work area

In this area you can drag and drop the components. See Build up the page on page 71 for more information about building up a Pagebuilder page using components.

## C: the topbar

The topbar functions.

- VARIABLES: when working with pages that use variable data, this area allows you to connect to a database collection, a data connector or a workflow object to test your page. See Working with Page Variables for more information.
- STYLE:
  - Select a styling from the drop-down list to apply a general styling. This styling will not overwrite the styling of the components.
  - Select **STYLE** to create a new styling.

See Styling of the HTML page for more information on HTML page styling in Page Builder.

• VIEW: select this button to preview the page.

- **DUPLICATE**: select this button to duplicate the page.
- NEW: select this button to create a new HTLM webpage with PAGEBUILDER.
- SAVE: select this button to save the current page.

**Note:** The system will automatically save the webpage every 30 seconds.

Web Page Settings (a). See Web Page Settings on page 69 for an overview.

#### D: the component configuration

In this area you can configure the parameters of the selected component.

#### E: the title of the Pagebuilder page

You can edit the title by double clicking.

#### F: MARS App

If your Pagebuilder page is a part of a MARS App, Pagebuilder shows this tab. If you open it, the assets and the workflows that are part of the MARS app are displayed.

# Web Page Settings

In Web Page Settings you can define the settings for the Pagebuilder page.

Select to open the Web Page Settings. The following options are available:

- Title: here you can enter a new title.
- Error page url: here you can enter a URL to redirect to in case the page could not be loaded.
- Apply style: if you select the checkbox, the selected style is displayed in the editor. If you leave this checkbox unselected, the applied style is hidden and the default style is displayed. However, the selected style will be applied to your HTML page.
- **Translations**: if you select the checkbox next to **Enable custom translations**, the text that you can enter in the text panel can be translated in the page.

**Note:** This checkbox is only available if your web page is part of a MARS app.

- Supported languages: here you can define the languages you want to support for translation.
  - **Note:** This checkbox is only available if your web page is part of a MARS app.
  - If you select **Export to XLIFFs**, you can select a folder where an XLIFF file will be created. This XLIFF file contains both the source and target language and can be translated using an XLIFF editor or translation tool. If

an XLIFF already exists that has the same name and the same language combination as the XLIFF file that you want to export, the XLIFF files will be merged.

• If you select **Import from XLIFFs**, you can select a folder where the translated XLIFF files are located to import them. Make sure that the XLIFF files contain the identical language code in its name, this language should be the same as in the original exported XLIFF file.

#### Example 1:

You want to translate the Submit button in a Pagebuilder form from English to Spanish and Italian.

+	R newpage.					查 VARIABLES	STYLE     Default Style * 0	Þ v	IEW C	2 -
٩									Form	
Cloud The white t logo	flow topbar opbar with cloudflow	*							Work	flow*
Title bark bl The dark bl black go ba	ar lue topbar with the ack button	Fill	in the form	ו						nput-
A Title A simple te	xt title		c	Job id*	type here			I	Subr	hitted ssage
A Text pa An extende	anel id text panel			Туре	Prepress     Production			I		
A simple in	nage	Upload	Cloudflow					I	S	ubmit
<b>∓</b> Naviga The cloudfi	tion ow navigation bar							I	buttor	n text
III Table A generic ta	able							I	S	ubmit
I Job tal	ble d table for listing jobs							I	button la	ayout
Draw differ	ent tabs							ŀ	Actions	
Form An input for	rm to submit to a Kiosk"	Browse	fies						<ul> <li>Sub</li> <li>Car</li> </ul>	mitted
Button Ageneric b	utton				Submit					
☆ Custor A reference	to a custom page	_								

Follow these steps:

- 1. Open the Web Page Settings and select the checkbox next to Enable custom translations.
- Select the checkboxes next to español and Italiano. They will be the target languages. The selected CLOUDFLOW UI language in SETTINGS > SETTINGS > INTERNATIONAL > LANGUAGE is English, so the system automatically selects English as the source language.
- 3. Save the Web Page Settings.
- 4. Select the yellow icon in the field Submit button text:

0



5. Enter Submit in the English field.

Translate: Submit button	text of Form	8
English:	Submit	Â
español:		
Italiano:		
		Cancel Save

- 6. If you know the Spanish and Italian translation for Submit, enter the translations and select Save.
- 7. The yellow icon has become green in the page:

Submit	0

- **Note:** The translations are stored in a \*.json file in a subfolder in the assets folder of the MARS App.
- 8. If you don't know the Spanish and Italian translations for **Submit**, leave the translation fields empty and select **Save**.
- 9. Open the Web Page Settings and select Export to XLIFFs.

10. Select the folder where you want to store the XLIFF files. In that folder, two XLIFF files are created:

- One that contains English as the source language and Spanish as the target language.
- One that contains English as the source language and Italian as the target language.
- 11. You can translate these XLIFF files in an XLIFF tool, or send the files to a translator.
- 12. When the XLIFF files are ready, store them in a folder. Make sure that the files are called xxx\_es.xliff and xxx\_it.xliff.
- 13. Open the Web Page Settings and select Import from XLIFFs.
- 14. Select the folder where you have stored the XLIFF files to import the translations.
- **15.** If you switch your CLOUDFLOW system to one of the two translated languages, you will see that the Submit button now has a Spanish or Italian translation.
- Attention: To prevent issues with the translation, you cannot duplicate Pagebuilder pages (for example with an API call). Each Pagebuilder page has a unique id, and each component has its own global unique id. Translations refer to those unique ids. Duplicating a Pagebuilder page results in two pages in which the translations are linked to the same translation file. If you edit a translation in the duplicated file, it will also change in the original file.

# Build up the page

To build up the page with the components, take the following steps:

1. Drag the components to the empty component fields in the work area:

Q Cloudflow topbar	CLOUDFLOW	н
The white topbar with cloudflow logo	UNUNUNUNUNUNUN	
Title bar The dark blue topbar with the black go back button		
A Title A simple text title		

2. When the component is placed on the page, hover over the component for extra options:



- A: with these options you can move, add or delete component fields. A component field is a placeholder where the component is placed into.
  - 🔊: allows you to change the component field's position by dragging and dropping it to another location on the page.
  - **+**: allows you to add a component field.
  - 💼: allows you to delete the component field.
- **B**: with these options you can add or delete a component.

  - 🛍: allows you to delete the component.
- C: with this option you can change the width of a component by dragging the arrow to the desired width. By decreasing the width of a component in the work area, you can add another one on the remaining space on the same field.
- **3.** Configure the components in the work area. See Pagebuilder components on page 72 for a complete overview of the available components.

# Pagebuilder components

This is an overview of the available components in PAGEBUILDER.

# **CLOUDFLOW** topbar

The CLOUDFLOW topbar adds the white topbar with the CLOUDFLOW logo.

CLOUDFLOW	HELLO <ful< th=""><th>LNA</th></ful<>	LNA
-----------	---------------------------------------	-----

It has the following configuration options:

- **Icon**: here you can enter the link or browse to the icon you want to use in the topbar. The icon needs to be available on a webserver. If you select the icon from within a tab in CLOUDFLOW, you are redirected to the HOME tab.
- Title: here you can enter the title that appears in the top bar.
- Scope: if you select the checkbox Show scope selector, the scope selector will be available in the webpage.
- User menu: if you select the checkbox Show user menu, the user menu will be available in the webpage.
- Style: here you can define the style of the topbar. Options:
  - Cloudflow style: if you select this option, the style of the topbar will match the CLOUDFLOW style.
  - Global page style: if you select this option, the style of the topbar will match the style you have selected in STYLE on top of the page.

#### Title bar

The Title bar component adds the dark blue topbar with the black back button.

TITLE BAR

It has the following configuration options:

• Icon: here you can enter the link or browse to the activity icon you want to use in the title bar. The icon needs to be available on a webserver.
- Title: here you can enter the title that appears in the title bar.
- **Buttons**: here you add buttons that appear at the right of the title bar. Select *to open the Manage buttons* tab. Each column represents a button with its icon and text.
  - Select + or Add column to add a column.
  - Select  $\div$  to drag the column to another position.
  - Select to remove the column.
  - Select  $\boxtimes$  to close the tab.
  - Icon: here you can select an icon you want to use for the button.
  - Label: here you can enter the name that appears on the button.
  - Action name: here you can define a name for an action. The action name appears in the section Actions in the component settings. For each action name there is a dedicated action. See Working with Actions on page 104 for more information.
  - Tooltip: here you can enter a tooltip that appears when hovering over the button.

Select Save to save the configuration.

# Actions

With **Actions** you can manage actions that occur after a trigger. If you have configured one or more button actions, there is a dedicated action for each button:

lanage buttons		+ Add column 🛛 🙁	
Icon x • Label Close Action name* Close Tooltip Click to close	+ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓		Actions <ul> <li>action: Continue</li> <li>action: Close</li> </ul>

- A: Configured action 1
- **B**: Configured action 2
- C: Available actions. If you define a trigger, an action will start. See Working with Actions on page 104 for more information on Actions.

**Title** The **Title** component adds a title or a simple text line.

# Title

It has the following configuration options:

- Text: here you can enter the text content.
- Type of title: here you can select the type of text. Options are Title, Subtitle or Text.
- Text align: here you can select the text alignment. Options are Auto, Left, Center or Right.

# Style

- Background Color: here you can select the background color of the text block.
- Font Color: here you can select the font color.

# Actions

With Actions you can manage actions that are started by a trigger. The following triggers are available:

- Click: this trigger will start an action if you click the title or text.
- Double click: this trigger will start an action if you double click the title or text.

See Working with Actions on page 104 for more information on Actions.

# Text panel

The Text panel component adds a text panel with advanced text editing possibilities.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad min laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pa proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

It has the following configuration options:

• Edit: here you can manage the content and the style of the text. Select ✓ to open the text panel, which provides advanced text editing possibilities. If there are page variables present in your page, you can use these variables by selecting Variables on the top right.

If your web page is part of a MARS app, you can add translation strings. See Web Page Settings on page 69 for more information.

# Style

- Background Color: here you can select the background color of the text block.
- Font Color: here you can select the font color.

# Image

This Image component adds an image.



It has the following configuration options:

- Url of image: here you can browse to the image. The image needs to be available on a webserver. You can use a relative path.
  - **Note:** If you want to create an image path from a variable, activate the Page Variables and ALT+ click on the image URL.
- Width: here you can select the image width.
- Height: here you can select the image height.
  - **Note:** You can define both the image width and height in:
    - Pixels (for example 100px).
    - Percentage (for example 100%). In this case, the image will scale together with the HTML page size.
- Horizontal Align: here you can select the horizontal alignment of the image. Options are Left, Center and Right.

# Style

• Background color: here you can select the background color.

# Actions

With Actions you can manage actions that are started by a trigger. The following triggers are available:

- Click: this trigger will start an action if you click the title or text.
- **Double click**: this trigger will start an action if you double click the image.

See Working with Actions on page 104 for more information on Actions.

# Link

The Link component provides a clickable link to open another web page.

http://www.google.com

The component has the following configuration options:

- URL: here you can define the URL of the web page where you will be directed to. You can use variables to build up the URL. Select the pencil to open Expression Builder.
- Target: here you can specify if the web page should be opened in the same tab or in another tab.
- Content Text: here you can specify the content text, which is the visible part of the link.
- Style: here you can define the style of the link.

#### Navigation

The Navigation component adds the CLOUDFLOW navigation bar.



It has the following configuration options:

Buttons: here you can define the navigation buttons. Select 🖋 to open the Manage buttons tab.

Each column represents a button with its icon and text on the navigation bar.

- Select + or Add column to add a column.
- Select rightarrow to drag the column to another position.
- Select to remove a column.
- Select Select to close the Manage buttons tab.
- Icon: here you can select the icon that you want to use for the button.
- Label: here you can enter the name that appears on the button.
- Link: here you can enter the link of the page that opens after selecting the button.
- Tooltip: here you can enter a tooltip that appears when hovering over the button.

Select Save to save the configuration.

• Active link (optional): here you can specify the matching URL of the page to highlight the navigation button. If you don't specify a URL, Pagebuilder will automatically specify the navigation button to highlight, based on the URL.

#### Style

• Background Color: here you can select the background color.

# Table

The Table component adds a generic table. It reads the data from the database and displays it in the table.

Name	•	Country	Extra Info	
Customer A		NL	blocked	
Customer B		BE	beta custon	ner
Customer C		BE		
Customer D		BE	beta custon	ner
		4		

The component has the following configuration options:

- **Collection**: here you can select the database collection from which the data is read and copied to the table. Options:
  - **Custom Objects**: if you select one of the custom objects collections, the data from this collection in the database is read. See Using custom objects in the database on page 466 for more information on custom objects.
  - The Jobs collection: if you select this option, the data from the Jobs collection in the database is read.
    - **Note:** Job table is a dedicated component for reading Job tables.
    - **Note:** CLOUDFLOW returns recursive Job data. This means that Job information of nested Jobs are also returned.
  - Start from DBIO: if you select this option, the data is collected from a workable that was created by a workflow that starts with the Start from DBIO and sets a reply with the Set DBIO Reply.
    - **Note:** In case you want to use filters in a generic table that is driven by a workflow containing a **Start From DBIO** node, the returning interface of HTTP Reply node must be a parameter **columns** with an array containing objects with a column interface. A column must contain a **name** as a string and a **data\_type** as a string, number or boolean.
    - Item ID: here you can define the key name of an item in the collection.
  - Data Connector Data Link : if you select this option, the DATALINK connections set up on your CLOUDFLOW are connected to read the data.
    - Item ID: here you can define the key name of an item in the collection.
  - \*Define own list\*: if you select this option, your own defined data object is used in the table.
    - **Note:** You can only use this option in case of one fixed list in a record (for example separations or a fixed URL list).
    - List: here you can define the array with objects to use in the table.
    - Item ID: here you can identify the name of the unique key that identifies each item (row in the table). This key's value is what will be passed for certain actions to identify the row(s) selected.
  - File: if you select this option, the data from a JSON file is read.
    - Data file url: here you can define the URL to the JSON file you want to read the data from.
    - Item ID: here you can define the key name of an item in the collection.
  - **Note:** you can use dynamic collections. Use the following syntax:
    - Jobs: nucleus.job
    - Custom object: nucleus.customobjects.<name of collection>
    - Dataconnector: dataconnector.<name>/
    - Start From DBIO: whitepaperCRUD.<name whitepaper>/<input name>

Do not use dots (.) or slashes (/)

- Columns: here you can manage the table columns. Select 🖋 to open the Manage columns tab.
  - Select + or Add column to add a column.
  - Select 

     to drag the column to another position.
  - Select to remove the column.
  - Select 🗵 to close the tab.
  - **Type**: here you can select the column type. Depending on the type you select, the sort and search options will differ. Options:
    - text: if you select this type, the column content will be interpreted as text.
    - styled text: if you select this type, the column content will be interpreted as styled text.
    - number: if you select this type, the column content will be interpreted as a number.
    - image: if you select this type, the column content will be interpreted as an image. Extra options:
      - Max width: here you can select the maximum width of the image.
      - Max height: here you can select the maximum height of the image.
    - **true/false**: if you select this type, the column content will be interpreted as a boolean. See **mapping** type for more boolean functionalities.
    - date (ISO-format): if you select this type, the column content will be interpreted as a date. Extra options:
      - **Display format**: here you can define if you want to display the local date only or the local date and time.
    - link: if you select this type, the column content will be interpreted as a link. Extra options:
      - Link label format: here you can define the label of the link.
    - unit: if you select this type, the column content will be interpreted as a unit. Extra options:
      - Unit type: here you can select the unit type (for example length, resolution, scaling...
      - Storage unit: here you can select the storage symbol.
    - mapping: if you select this type, you can create a fixed list with column items and map them in order to

display them as an image, an icon or text in the column. Select 🖋 at the bottom to define the fixed list and

the mappings. For each value you want to map, you have to add a mapping configuration (

- Type of the values: here you can select the type of the values in the fixed list.
- Compare: here you can enter the value you want to map.
- Display: here you can select the type of value to which you want to map. Type options:
  - **Icon**: if you select this option, the predefined value will be mapped to and displayed as an icon. You can select the icon in the icon list.
  - **Image**: if you select this option, the predefined value will be mapped to and displayed as an image. You can define the image URL in the text field and you can define the maximum width and height.
  - Text: if you select this option, the predefined value will be mapped to and displayed as text. You can enter the text in the text field.
- Selector: here you can enter text in a text field that will be displayed in case the column content is searchable.

#### Example

You have table with a column **Status**, where you want to display the print status of a file. The values of that column are retrieved from the database as a boolean (**true** or **false**) from a field called **print**. In your table you want to map the boolean value **true** to an icon. To do so, take the following steps:

- Add a column to your table with **mapping** as the column **Type**, **Print Status** as the **Title** and **print** as the **Key (path)**.
- Select 🖋 to open the mapping configuration.
- Select **true**/**false** as the value type.

- Select icon as the Display type and select the icon you want to use to display the true values, for example the Printer icon (=)
- If needed, enter a text in the text field that will be used in the file selector (for example All files that are ready to print). In that case, make sure to select the checkbox Searchable.

The files that are ready to print are now displayed in the column **Print Status** with the icon  $\bigoplus$ . You can also filter on all the rows that contain **All files that are ready to print**.

- **next state buttons**: if you select this type, a column with the next state buttons of a Job is present. If next states are set to a Job (via the **Update Job** node), these will be visible in this column. If you select one of the states, this state will be set to the Job.
- Title: here you can enter the title of the column that is displayed on the column header.
- Key (path): here you can define the key of the database collection that you want to display in the column. If you start typing, a list with one or more auto fill suggestions appears. Select the key you want to use.
- Sortable: if you select this checkbox, you can sort the content in the column.
  - **Note:** If you have selected **mapping** as the **Type**, a message appears to warn the users that the sorting is based on the key value and not on what is displayed.
- Searchable: if you select this checkbox, a search field appears and you can search the content in the column.
- **Default search**: here you can specify a default search string.
- Width: here you can select the column width.

#### Example

You have a **customobjects** database collection that contains the collection **customers** (customobjects.customers). This collection contains a record for each customer, containing the fields **\_id**, **name**, **info**, **country** and **address**( for all of which the dot notation is **customobjects.customers.X**. For example, customobjects.customers.name).

The content of customobjects.customers is this:

```
[
{
    " id" : ObjectId("5a676619c60c00ad0000005"),
    "name" : "Customer A",
    "info" : "blocked",
    "country" : "NL"
    "address" : "Tulpstraat 1\nGent"
},
{
    " id" : ObjectId("5a6766f3ea0c00e30000005"),
    "name" : "Customer B",
    "info" : "beta customer",
    "country" : "BE"
    "address" : "Klaproosstraat 1\nGent"
},
{
    " id" : ObjectId("5aa119348761655a34d73a53"),
    "name" : "Customer C",
    "info" : "",
"country" : "BE"
    "address" : "Roosstraat 1\nGent"
},
{
    " id" : ObjectId("5aa129a98761655a34d73a57"),
    "name" : "Customer D",
    "info" : "beta customer",
    "country" : "BE"
    "address" : "Leliestraat 1\nGent"
}
```

÷ • +	÷ •+	+ <b>O+</b>	
Туре	Туре	Туре	
text 🗘	text 🗘	text 🗘	
Title	Title	Title	
Name	Country	Extra Info	
Key (path)	Key (path)	Key (path)	
name	country	info	
Sortable	Sortable	Sortable	
Searchable	Searchable	Searchable	
Width	Width	Width	
Auto 🗘	Auto 🗘	Auto 💠	
type specific options	type specific options	type specific options	

way:

]

**Note:** In scenarios where the array does not contain objects (for example an array of strings), CLOUDFLOW will generate a key named **value** that will refer to each element in the array. Use the key named **value** in these circumstances if no unique id is available or needed.

Filter: here you can manage filters for the database content that should be displayed. Select 🖍 to open the tab where you can manage the filters.

1. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.

You want to create a table that contains three columns: one that displays the name of the customers, one that displays the country and one that displays

- 2. Select the criteria.
- 3. If you want to add a rule for a combined filter, select AND or OR and define the additional rules or groups.
- 4. If needed, select **Delete** to delete a rule or a group of rules.
- 5. Select Save.

#### Example

Have a look at the previous example with the database collection customobjects.customers. If you want the column only to display the customers that are blocked, you can use this filter:

info equal blocked

If you want to display the blocked customers from Belgium, you can use:

```
info equal blocked
AND
country equal BE
```

╤

F

Note: Filter is not used with the \*Define own list\* collection.

**Note:** If you want to use the filter on a boolean, select (custom), enter the key value in the first field, followed by equal and true or false.

(custom)	selected	equal	true
(custom)	Selected	- cquai	lide

In this case, the filter will filter on all keys of which the value is true.

**Topbar buttons**: here you can define buttons that appear at the top of the table. Select do open the **Manage buttons** dialog. Each column represents a button with its icon and text above the table.

- Select + or Add column to add a column.
- Select 💠 to drag the column to another position.
- Select to remove the column.
- Select  $\boxtimes$  to close the tab.
- Icon: here you can select an icon you want to use for the button.
- Label: here you can enter the name that appears on the button.
- Action name: here you can define a name for an action. The action name appears in the section Actions in the component settings. For each action name there is a dedicated action. See Working with Actions on page 104 for more information.
- Enabled when: here you can define when the button is available and the action can be triggered. Options:
  - Always: the button is always enabled.
  - No selection: the button is enabled if no table row is selected.
  - Only one selected: the button is enabled if only one table row is selected.
  - One or more selected: the button is enabled if one or more table rows is selected.
  - Multiple selected: the button is enabled if two or more table rows are selected.
  - **Conditional multiple**: the button is enabled if one or multiple rows are selected that match a conditional query. Select **\*** to open the tab to create the rule(s) for the conditional query.
  - Conditional only one: the button is enabled if one row is selected that matches a conditional query. Select
     to open the tab to create the rule(s) for the conditional query.
- Tooltip: here you can enter a tooltip that appears when hovering over the button.

Select **Save** to save the configuration.

- Sort by: here you can select the key on which you want to sort the columns.
- **Order**: here you can define the sort order.

- **Row conditional styling**: here you can define style classes for each row based on conditions. Select  $\checkmark$  to open the dialog where you can manage the conditional queries. To define query and a matching Style, follow these steps:
  - 1. Define the Conditional query:
    - a. Select the key from the drop-down list and select the matching criteria.
    - **b.** If you want to add a second rule or group, select **AND** or **OR**. A rule is a single rule, a group is a group of single rules.
  - 2. Select the Style to apply to this query.
  - **3.** Select **•** to add define another **Conditional query** and **Style**.

Conditional query	Style
AND OR Add rule 🖬 Add group	
name 🗢 equal 🗢 Customer A 🗢 Delete	Prima
AND OR Add rule S Add group	
name     Image: equal     Image: Customer B	Secor
	Conditional query         AND OR         name © equal © Customer A         OR         OR         Pelete         AND OR         © Add rule © Add group         Image: Customer B

In this example, the Primary style will be applied to all to table rows with a key **name** that is equal to **Customer A**, and the Secondary Style will be applied to all table rows with a key **name** that is equal to **Customer B**.

- **Disclosure**: here you can define options regarding the disclosure.
  - **Disclosure button**: if you select this checkbox, a disclosure button is added. This disclosure button can be expanded or collapsed to show more details about the row element.
  - **Disclosure content**: here you can select the component that will be displayed when the disclosure button is expanded.

In your table, you want to add a disclosure button to display the customer's address. In that case, you have to create a new Pagebuilder file that contains the address for each customer. To do so, follow these steps:

- 1. Create a new Pagebuilder file called **address.html**.
- 2. Add a Title component to the page and select Text as the Type of title.
- **3.** In **VARIABLES**, connect to your collection **customers** in the **customobjects** database collection. This way, you load Page Variables in your page. See Working with Page Variables on page 108 for more information on Page Variables.
- 4. Since you have loaded Page Variables in your page, the pencil to open Expression Builder appears next to the **Text** component settings:

Text	title_text	ø	
------	------------	---	--

Select the pencil to open Expression Builder.

5. In Expression Builder, select Page Variables > address and drag address to the area where you can build the expressions. Select Save:

Title: Text		
Q. Search	Expression 🕢 Value type: As is	Origin
Page country Page Variables Info Custom Variables	address	
	Variables	
Add Variable	address: Tulpstraat 1Gent	
	Result Tulpstraat 1 Gent	
		Cancel

The page that contains the address is now ready. The next step is to load it the table. To do so, follow these steps:

- 1. Open the page where you have created the table.
- 2. Select the checkbox next to add disclosure button.
- 3. In Disclosure content, select to the page you have just created (address.html).
- 4. This is the result:

	Name	Country		Extra Info
•	Customer A	NL		blocked
	Tulpstraat 1 Gent			
۲	Customer B	BE		beta customer
•	Customer C	BE		
	Roosstraat 1 Gent			
•	Customer D	BE	4	beta customer

- **Input type**: here you can select the input type that sends data to the disclosure content. Options:
  - **default**: if you select this option, the default input type is selected.
  - **workflow**: if you select this option, you can specify a workflow to input data in the disclosure content. In this case, a workflow is started when the disclosure content is revealed. The variables generated by the workable can be used in the disclosure content.
  - Workflow\*: here you can select the workflow to start when the disclosure content is revealed. You need to select a workflow that starts with a **Start From Web Request** node.
  - Input\*: here you can select the input of the workflow.
  - **Open workflow:** if you select this link, the workflow that you specified is opened in the Workflow Editor.
- Only one disclosure panel open: if you select this checkbox, only one disclosure panel can be opened at the same time.
- Header: here you can define options regarding the header.

=

- **stick to the top**: if you select this checkbox, the header of the table sticks to the top of the browser window when scrolling through a long table.
- Maximum rows: here you can select the maximum amount of rows that is displayed in the table.
- **show page and amount of rows**: if you select this checkbox, the page and the number of rows are displayed in the table.
- **Style paging buttons**: here you can select the style of the paging buttons from the drop-down list. See Styling on page 113 for more information about Pagebuilder styles.
- **user configuration**: if you select this checkbox, a cogwheel appears when you preview the page with the **VIEW** option and when you hover over the right side of the table header. When you select the cogwheel, the user configuration opens where you can show or hide table columns by selecting or deselecting the columns.
- **Refreshing delay**: here you can select the delay after which the page is automatically refreshed. Select **No** refreshing if you don't want automatic refreshing of the page.
- Style: here you can select the style. See Styling on page 113 for more information about Pagebuilder styles.

**Note:** The general style will not overwrite the style of a component.

# Style

- Background Color: here you can select the background color.
- Font Color: here you can select the font color.

#### Actions

With Actions you can manage actions that occur after a trigger. The following triggers are available:

- Row click: this trigger will start an action if you click the row.
- Double Row click:: this trigger will start an action if you double click the row.
- **action: Image click:** this trigger will start an action if you click an image in the table row. This trigger is only available if an image column is defined in the table **Columns**:

Manage columns		+ Add column
÷ • +	÷ •+	
Туре	Туре	
text 🛟	image 🛟	
Title	Title	1
Customer name	Image	
Key (path)	Key (path)	
cusname	imagekey	
Sortable	Sortable	
Searchable	Searchable	
Width	Max width	
Auto 🗘		
type specific options	Max height	
		Cancel

• **Topbar: actionName**: this trigger will start an action if you click a button in the table Topbar. This trigger is only available if the button is defined in the **Topbar** buttons:

Manage buttons	+ Add column
	Cancel Sav

See Working with Actions on page 104 for more information on Actions.

# Job Table

The Job Table component adds a dedicated Job table. It reads the data from the database and displays it in the table.

identifier	description	name	state			modification -	Label
						start filtering	
	job_test	4546456	state1	Û	۲	10/14/2019 3:09 PM	
774435		774435 -	Waiting for files	Û	۲	05/24/2019 11:52 AM	
4546456		4546456	Waiting for files	Û	۲	05/24/2019 11:52 AM	Files arrived
643434343		643434343	Waiting for files	Û	۲	05/24/2019 11:24 AM	Files arrived
4546456		4546456 -	New order	Û	۲	05/23/2019 3:16 PM	
545546		545546	Waiting for files	Û	۲	05/23/2019 3:16 PM	Files arrived
45456		45456	Waiting for files	Û	۲	05/23/2019 3:16 PM	Files arrived
963		963	Waiting for files	Û	۲	05/23/2019 3:14 PM	Files arrived
456		456 -	Waiting for files	Û	۲	05/23/2019 3:14 PM	
345		345 -	New order	Û	۲	05/23/2019 2:57 PM	
789		789	Waiting for files	Û	۲	05/23/2019 2:52 PM	Files arrived
456		456	Waiting for files	Û	۲	05/23/2019 2:51 PM	Files arrived
345		345	Waiting for files	Û	۲	05/23/2019 2:29 PM	Files arrived
234		234 -	New order	Û	۲	05/23/2019 2:26 PM	
234		234	Waiting for files	Û	۲	05/23/2019 2:26 PM	Files arrived
123		123 -	New order	Û	۲	05/23/2019 1:29 PM	
item123		item123	Waiting for files	Û	۲	05/23/2019 1:29 PM	Files arrived
EA0204411		P100131.1	700 Ready for Production	ŧ	۲	03/19/2019 3:33 PM	
39ecba8f- 1078-4d20- 8cb4- 258fab41cd7d		EA0204411_back.pdf		÷	۲	03/19/2019 3:33 PM	

**Note:** CLOUDFLOW returns recursive Job data. This means that Job information of nested Jobs are also returned.

The component has the following configuration options:

- **Columns**: here you can manage the Job table columns. Select *start to open the Manage columns* tab.
  - Select + to add a column.

  - Select to remove the column.
  - Select store the tab.
  - **type**: here you can select the column type. Depending on the type you select, the sort and search options will differ. Options:
    - text: if you select this type, the column content will be interpreted as text.
    - styled text: if you select this type, the column content will be interpreted as styled text.
    - number: if you select this type, the column content will be interpreted as a number.
    - image: if you select this type, the column content will be interpreted as an image. Extra options:
      - Max width: here you can select the maximum width of the image.
      - Max height: here you can select the maximum height of the image.
    - **true/false**: if you select this type, the column content will be interpreted as a boolean. See **mapping** type for more boolean functionalities.
    - date (ISO-format): if you select this type, the column content will be interpreted as a date. Extra options:
      - **Display format**: here you can define if you want to display the local date only or the local date and time.
    - link: if you select this type, the column content will be interpreted as a link. Extra options:
      - Link label format: here you can define the label of the link.
    - unit: if you select this type, the column content will be interpreted as a unit. Extra options:
      - Unit type: here you can select the unit type (for example length, resolution, scaling...
      - Storage unit: here you can select the storage symbol.
    - mapping: if you select this type, you can create a fixed list with column items and map them in order to display them as an image, an icon or text in the column. Select at the bottom to define the fixed list and the mappings. For each value you want to map, you have to add a mapping configuration (...).
      - Type of the values: here you can select the type of the values in the fixed list.
      - **Compare**: here you can enter the value you want to map.

Compare options are:

- For numbers : 'less than or equal', 'less than', 'equal to', 'greater than', 'greater than or equal'.
- For text: 'equal to', 'contains text like', 'begins with', 'ends with'.
- Display: here you can select the type of value to which you want to map. Type options:
  - **Icon**: if you select this option, the predefined value will be mapped to and displayed as an icon. You can select the icon in the icon list.
  - **Image**: if you select this option, the predefined value will be mapped to and displayed as an image. You can define the image URL in the text field and you can define the maximum width and height.
  - Text: if you select this option, the predefined value will be mapped to and displayed as text. You can enter the text in the text field.
- Style Class: here you can enter the style you want to map.
- Selector: here you can enter text in a text field that will be displayed in case the column content is searchable.
- Action: here you can enter the actions that will be referenced to.

#### Example

You have table with a column **Status**, where you want to display the print status of a file. The values of that column are retrieved from the database as a boolean (**true** or **false**) from a field called **print**. In your table you want to map the boolean value **true** to an icon. To do so, take the following steps:

- Add a column to your table with **mapping** as the column **Type**, **Print Status** as the **Title** and **print** as the **Key (path)**.
- Select 🖉 to open the mapping configuration.
- Select **true**/**false** as the value type.
- Select icon as the Display type and select the icon you want to use to display the true values, for example the Printer icon (=)
- If needed, enter a text in the text field that will be used in the file selector (for example All files that are ready to print). In that case, make sure to select the checkbox Searchable.

The files that are ready to print are now displayed in the column **Print Status** with the icon  $\bigoplus$ . You can also filter on all the rows that contain **All files that are ready to print**.

- view: if you select this type, a column with the view icon (1) is present. Selecting this icon can trigger an action. See View click for more information.
- **remove**: if you select this type, a column with the remove icon is present. If you select this icon, the Job will be removed from the table.
- **next state buttons**: if you select this type, a column with the next state buttons is present. If next states are set to a Job (via the **Update Job** node), these will be visible in this column. If you select one of the states, this state will be set to the Job.
- title: here you can enter the title of the column that is displayed on the column header.
- **key (path)**: here you can define the key of the Jobs collection in the database that you want to display in the column. If you start typing, a list with one or more auto fill suggestions appears. Select the key you want to use.

#### Example

You have the following Job in the database:

If you want to display a column with the Job names, you have to enter name in the field key (path).

- sortable: if you select this checkbox, you can sort the content in the column.
- searchable: if you select this checkbox, a search field appears and you can search the content in the column.
- Default search: here you can specify a default search string.
- width: here you can select the column width.

Filter: here you can manage filters for the Jobs content that should be displayed. Select 🖋 to open the tab where you can manage the filters.

- 1. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.
- **2.** Select the criteria.
- 3. If you want to add a rule for a combined filter, select AND or OR and define the additional rules or groups.
- 4. If needed, select **Delete** to delete a rule or a group of rules.
- 5. Select Save.

#### Example

If you want the column only to display the Jobs of which the identifier start with 123, you can use this filter:

```
identifier begins with 123
```

If you want to display the Jobs of which the identifier start with 123 and with state For approval, you can use:

```
identifier begins with 123
AND
state equal For approval
```

- **Topbar buttons**: here you can define buttons that appear at the top of the table. Select *s* to open the **Manage buttons** dialog. Each column represents a button with its icon and text above the table.
  - Select + or Add column to add a column.
  - Select rightarrow to drag the column to another position.
  - Select to remove the column.
  - Select 🗵 to close the tab.
  - Icon: here you can select an icon you want to use for the button.
  - Label: here you can enter the name that appears on the button.
  - Action name: here you can define a name for an action. The action name appears in the section Actions in the component settings. For each action name there is a dedicated action. See Working with Actions on page 104 for more information.
  - Enabled when: here you can define when the button is available and the action can be triggered. Options:
    - Always: the button is always enabled.
    - No selection: the button is enabled if no table row is selected.
    - Only one selected: the button is enabled if only one table row is selected.
    - One or more selected: the button is enabled if one or more table rows is selected.
    - Multiple selected: the button is enabled if two or more table rows are selected.
    - **Conditional multiple**: the button is enabled if one or multiple rows are selected that match a conditional query. Select **\*** to open the tab to create the rule(s) for the conditional query.
    - Conditional only one: the button is enabled if one row is selected that matches a conditional query. Select
       to open the tab to create the rule(s) for the conditional query.
  - Tooltip: here you can enter a tooltip that appears when hovering over the button.

Select Save to save the configuration.

- Sort by: here you can select the key on which you want to sort the columns.
- Order: here you can define the sort order.
- Save sort by and order: if you select this checkbox, the sort order of the columns is saved after refreshing the page.

- **Row conditional styling**: here you can define style classes for each row based on conditions. Select ✓ to open the dialog where you can manage the conditional queries. To define query and a matching Style, follow these steps:
  - 1. Define the Conditional query:
    - **a.** Select the key from the drop-down list and select the matching criteria.
    - **b.** If you want to add a second rule or group, select **AND** or **OR**. A rule is a single rule, a group is a group of single rules.
  - 2. Select the **Style** to apply to this query.
  - **3.** Select **•** to add define another **Conditional query** and **Style**.

=

Conditional query				Style
AND OR			Add rule 🖪 Add group	
state	ᅌ 🛛 equal	Ready for Production	O Delete	Primary

In this example, the Primary style will be applied to all to table rows with the key **state** that is equal to **Ready for Production**.

- **Disclosure**: here you can define options regarding the disclosure.
  - Add disclosure button: if you select this checkbox, a disclosure button is added. This disclosure button can be expanded or collapsed to show more details about the Job.
  - **Disclosure content**: here you can select the component that will be displayed when the disclosure button is expanded.
  - Input type: here you can select the input type that sends data to the disclosure content. Options:
    - **default**: if you select this option, the default input type is selected.
    - **workflow**: if you select this option, you can specify a workflow to input data in the disclosure content. In this case, a workflow is started when the disclosure content is revealed. The variables generated by the workable can be used in the disclosure content.
    - Workflow\*: here you can select the workflow to start when the disclosure content is revealed. You need to select a workflow that starts with a **Start From Web Request** node.
    - Input\*: here you can select the input of the workflow.
    - **Open workflow**: if you select this link, the workflow that you specified is opened in the Workflow Editor.
  - Only one disclosure panel open: if you select this checkbox, only one disclosure panel can be opened at the same time.
- Header: here you can define options regarding the header.
  - **stick to the top**: if you select this checkbox, the header of the Job table sticks to the top of the browser window when scrolling through a long table.
- Maximum rows: here you can select the maximum amount of rows that is displayed in the table.
- **show page and amount of rows**: if you select this checkbox, the page and the number of rows are displayed in the table.
- **user configuration**: if you select this checkbox, a cogwheel appears when you preview the page with the **VIEW** option and when you hover over the right side of the table header. When you select the cogwheel, the user configuration opens where you can show or hide table columns by selecting or deselecting the columns.
- **Style paging buttons**: here you can select the style of the paging buttons from the drop-down list. See Styling on page 113 for more information about Pagebuilder styles.
- **Refreshing delay**: here you can select the delay after which the page is automatically refreshed. Select **No** refreshing if you don't want automatic refreshing of the page.
- Style: here you can select the style. See Styling on page 113 for more information about Pagebuilder styles.

Note: The general style will not overwrite the style of a component.

# Style

- Background Color: here you can select the background color.
- Font Color: here you can select the font color.

#### Actions

With Actions you can manage actions that occur after a trigger. The following triggers are available:

- Row: this trigger will start an action if you click on a row in the Job list.
- Double click row: this trigger will start an action if you double click on a row in the Job list.
- Select: this trigger will start an action if you select and deselect a row in the Job list.
- View click: this trigger will start an action if you click (Job details) in the Jobs list.
- **action: Image click:** this trigger will start an action if you click an image in the Job table row. This trigger is only available if an image column is defined in the Job table **Columns**:

Manage columns		+ Add column	٥
÷ •+	÷ <b>o</b> +		
Туре	Туре		
text 🗘	image 🛟		
Title	Title		
Customer name	Image		
Key (path)	Key (path)		
cusname	imagekey		
Sortable	Sortable		
Searchable	Searchable		
Width	Max width		
Auto 🗘			
type specific options	Max height		
		Cancel Save	•

• **Topbar: actionName**: this trigger will start an action if you click a button in the Job table Topbar. This trigger is only available if the button is defined in the **Topbar** buttons:

Manage buttons	+ Add column	۲
Always \$		
	Cancel	

See Working with Actions on page 104 for more information on Actions.

# Tabs The Tabs component adds tabs.

CUSTOMERS BE CUST	OMERS NL	
Name		Info
Customer B		beta customer
Customer C		
Customer D		beta customer
		4 F

Each tab is a custom page component.

It has the following configuration options:

- Tabs: here you can manage the tabs. Select 🖋 to open the Manage tabs window.
- Select + or Add tab to add a tab.
- Select  $\oplus$  to drag the tab to another position.
- Select to remove the tab.
- Select  $\blacksquare$  to close the tab.
- Title: here you can select the title of the tab.
- Url: here you can define the URL of the custom page component you want to use as tab.
- **Pass variables into component**: if you select this checkbox, the page variables are passed to the custom page component that you have defined for the tab.
- **Height**: here you can define the height of the tab.

# **Form** The **Form** component adds an input form to submit to KIOSK.

Fill in the form			
		Customer*	type here
		Job id*	
		Туре	
Upload	Cloudflow		
Browse file	95		
			Submit

**Note:** You should only use this component in case you want to see and display a **form**. If you don't want to display a form (for example if you want to start a workflow with an Action (see Start Flow on page 107 for more information), use **Start From Web Request**.

The component has the following configuration options:

- Workflow: here you can select the workflow that contains the form you want add in the HTML page. The dropdown list contains:
  - All Jobs workflows, including the Standard Jobs Flow. See Job workflow on page 373 for more information on Jobs workflows.
  - All workflow that start with the **Start From KIOSK** node. The Start From KIOSK workflow can also do actions, for example passing information to a dialog or custom component.

• **Input**: here you can enter the Input name of the workflow that contains the form you want add in the HTML page. The input name is the name that is entered in the field **Name** in the **Start From Kiosk** node:

Manual Jobs (Start From Kiosk)	?	×
Name: Create Job		
Allow Access to Scopes:		
Enable for guest users		
Category: Jobs		
Kiosk parameters:		
Number of files: None		
Close	Sav	е

- **Open workflow** : if you select this option, the workflow will be added to the MARS App. This option is only available in case the Pagebuilder page is already part of a MARS App.
- Submitted message: here you can enter a message that will display after the asset is successfully submitted to the workflow.
- Note: If you have selected a connector in VARIABLES to retrieve the sample data to display, you can select ALT + click next to the fields Workflow, Input and Submitted message to show the pencil. Select the pencil to open Expression Builder.
- **Reset fields after submit**: if you select this checkbox, the form fields will be cleared after the asset is submitted to the workflow.
- Submit button text: here you can specify the text of the Submit button.
- Submit button layout: here you can define the style of the Submit button. Options:
  - Default: if you select this option, the layout of the button will be default.
  - Main type: if you select this option, the layout of the button will correspond with the Main button style that you have defined in the Styling. See Styling on page 113 for more information.
  - Second type: if you select this option, the layout of the button will correspond with the Second button style that you have defined in the Styling. See Styling on page 113 for more information.
- Show the cancel button: if you select this button, a Cancel button is displayed next to the Cancel button.

# Actions

With Actions you can manage actions that occur after a trigger. The following triggers are available:

- Submitted: this trigger will start an action if you have submitted an asset to the workflow.
- **Canceled**: this trigger will start an action if you have canceled.

See Working with Actions on page 104 for more information on Actions.

#### Button

The **Button** component adds a button.



It has the following configuration options:

• **Text**: here you can enter the text content.

- Icon: here you can select the button icon.
- Tooltip: here you can define a tooltip that appears when hovering over the button.
- Layout: here you can define the button layout. Options:
  - Default: if you select this option, the layout of the button will be default.
  - **Main type**: if you select this option, the layout of the button will correspond with the Main button style that you have defined in the Styling. See Styling on page 113 for more information.
  - Second type: if you select this option, the layout of the button will correspond with the Second button style that you have defined in the Styling. See Styling on page 113 for more information.

# Actions

With Actions you can manage actions that are started by a trigger. The following triggers are available:

- Click: this trigger will start an action if you click the button.
- **Double click**: this trigger will start an action if you double click the button.

See Working with Actions on page 104 for more information on Actions.

#### YouTube

The YouTube component adds a third party plugin, a YouTube video.

#### Options

100.0%	PACKZ	- WDP industria 1 and a Del	1400 1500 1600
19400		Checks	Share
	хатта и и и и и и и и и и и и и и и и и и	Number of Output Items: 1 Create: Since Fix Split document after page count: 1	
	CODO.		
104121	Tota		> OUT

It has the following configuration options:

• Video ID: here you can enter the ID of the YouTube video you want to embed. You can find this ID in the URL of the video page, right after the v=:

https://www.youtube.com/watch?v	=htD4ilFd7Vo	Ç
---------------------------------	--------------	---

- Width: here you can specify the width of the YouTube video.
- Height: here you can specify the height of the YouTube video.

#### Activate - Deactivate

When the slider for the plugin is deactivated **OO**, its components in Pagebuilder can still be part of the page layout

but will be grayed out and feature a blue button Accept YouTube ( Accept YouTube ). This must be clicked before its content is displayed.

# Twitter

The Twitter component adds a third party plugin, a Twitter timeline.

# Options



The component has the following configuration options:

- URL: here you can define the URL of the Twitter timeline. For example https://twitter.com/hybridsoftware.
- Width: here you can define the width of the component.
- Height: here you can define the height of the component.

# Activate - Deactivate

When the slider for the plugin is deactivated  $\bigcirc$ , its component in Pagebuilder can still be part of the page layout but it will be grayed out and feature a blue button **Accept Twitter**. This must be clicked before its content is displayed.

# Chart The Chart component adds a chart.



The component has the following configuration options:

- Type: here you can define the type of chart. Options:
  - Line: a line chart is a way of plotting data points on a line.
  - Horizontal bars: horizontal bars provide a way of showing data values represented as horizontal bars.
  - Vertical bars: vertical bars provide a way of showing data values represented as vertical bars.
  - **Pie** / **Doughnut**: pie and doughnut charts are divided into segments, the arc of each segment shows the proportional value of each piece of data.
  - Radar: a radar shows multiple data points and the variation between them.
  - **Pie radar**: a pie radar is a combination between the pie and radar charts.
- Width: here you can define the width of the chart.
- **Height**: here you can define the height of the chart.
- Source: here you can define the collection where the data is retrieved from. Options:
  - Custom Object: in this case, the chart data are retrieved from a Custom Object collection in the database.
  - Start from DBIO: in this case, the chart data are retrieved from the JSON data collected a workable that was created by a workflow that starts with the Start from DBIO and sets a reply with the Set DBIO Reply. The JSON must return an array containing numbers or an object with the next fields:
    - Value (required): the numeric value of the specific point.
    - Label: the label to show for that point. By default the text is empty.
    - Color (optional): the color of the point. By default the color is CLOUDFLOW blue.
    - Group (optional): the name of the group the point belongs to. The order of the points needs to be identical in all groups.
  - Data Connector Data Link : if you select this option, the DATALINK connections set up on your CLOUDFLOW are connected to read the data.
  - **Note:** you can use dynamic collections. Use the following syntax:
    - Custom object: nucleus.customobjects.<name of collection>
    - Dataconnector: dataconnector.<name>/
    - Start From DBIO: whitepaperCRUD.<name whitepaper>/<input name>

Do not use dots (.) or slashes (/)

- Filter: here you can manage filters for the content that should be displayed. Select 🖉 to open the tab where you can manage the filters.
  - 1. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.
  - **2.** Select the criteria.
  - 3. If you want to add a rule for a combined filter, select AND or OR and define the additional rules or groups.
  - 4. If needed, select **Delete** to delete a rule or a group of rules.
  - 5. Select Save.
- Show legend: if you select this checkbox, the legend is shown.
- Refreshing delay: here you can define the number of seconds after which the chart is refreshed.
- Default group name: here you can define the default group name.

# Style

• Background Color: here you can select the background color.

# Timeline

The Timeline component adds a gantt chart that displays multiple events on a timeline.

	May 2019								
	'hu 16	Fri 17	Sat 18	Sun 19	Mon 20	Tue 21	Wed 22	Thu 23	Fri 2
	Eve	nt one							
admins			Event two						
	Event three	Э							
user					E	Event five			
						E	Event four		
									+

The component has the following configuration options:

- Height (px): here you can define the height of the timeline.
- **Source**: here you can specify the source of the data in the timeline. The data needs to be an must be an array containing all the events as objects. Each event object is required to have the following structure:
  - \_id (string): the key to reference the event data.
  - **label** (string): the text that is displayed in the event.
  - from (ISO string | ISO string extended): the starting point of the event.
  - to (ISO String | ISO string extended): the ending point of the event.
  - group (string) (optional): to group multiple event with the same group name.
  - **color** (string) (optional): the color of the point. Default is CLOUDFLOW blue.

Options:

- Task list: if you select this option, the data are retrieved from a task\_list object in the database.
- Collection: if you select this option, the data are retrieved from:
  - A custom collection in the database.
  - The JSON data collected a workable that was created by a workflow that starts with the **Start from DBIO** and sets a reply with the **Set DBIO Reply**.

# If Source is set to Task list:

- Filter (only available here you can manage filters for the content that should be displayed. Select 🖉 to open the tab where you can manage the filters.
  - 1. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.
  - **2.** Select the criteria.
  - 3. If you want to add a rule for a combined filter, select AND or OR and define the additional rules or groups.
  - 4. If needed, select **Delete** to delete a rule or a group of rules.
  - 5. Select Save.

If **Source** is set to **Collection**:

- Collection: here you can define the collection where the data is retrieved from. Options:
  - Custom Object: in this case, the chart data are retrieved from a Custom Object collection in the database.
  - Start from DBIO: in this case, the chart data are retrieved from the JSON data collected a workable that was created by a workflow that starts with the Start from DBIO and sets a reply with the Set DBIO Reply.
- From parameter: here you can define the parameter name of the starting point value of the source.
- To parameter: here you can define the parameter name of the ending point value of the source.
- Selectable: if you select this checkbox, the item is selectable.

# Actions

With Actions you can manage actions that are started by a trigger. The following triggers are available:

- Item select: this trigger will start an action if you select the item. The checkbox Selectable needs to be selected.
- Item double click: this trigger will start an action if you double click the item.

See Working with Actions on page 104 for more information on Actions.

# **Custom Grid**

The Custom Grid component adds a custom grid.



The component has the following configuration options:

- Collection: here you can select the database collection from which the data is read and copied to the grid. Options:
  - **Custom Objects**: if you select one of the custom objects collections, the data from this collection in the database is read. See Using custom objects in the database on page 466 for more information on custom objects.
  - The **Jobs** collection: if you select this option, the data from the Jobs collection in the database is read.

**Note:** Job table is a dedicated component for reading Job tables.

- Start from DBIO: if you select this option, the data is collected from a workable that was created by a workflow that starts with the Start from DBIO and sets a reply with the Set DBIO Reply.
  - **For DBIO** node, the returning interface of HTTP Reply node must be a parameter **columns** with an array containing objects with a column interface. A column must contain a **name** as a string and a **data\_type** as a string, number or boolean.
  - Item ID: here you can define the key name of an item in the collection.
- Data Connector Data Link : if you select this option, the DATALINK connections set up on your CLOUDFLOW are connected to read the data.
  - Item ID: here you can define the key name of an item in the collection.
- File: if you select this option, the data from a JSON file is read.
  - Data file url: here you can define the URL to the JSON file you want to read the data from.
  - Item ID: here you can define the key name of an item in the collection.
- **Note:** you can use dynamic collections. Use the following syntax:
  - Jobs: nucleus.job
  - Custom object: nucleus.customobjects.<name of collection>
  - Dataconnector: dataconnector.<name>/
  - Start From DBIO: whitepaperCRUD.<name whitepaper>/<input name>

Do not use dots (.) or slashes (/)

Filter: here you can manage filters for the database content that should be displayed. Select 🖉 to open the tab where you can manage the filters.

- 1. Select Add rule or Add group. A rule is a single rule, a group is a group of single rules.
- 2. Select the criteria.
- 3. If you want to add a rule for a combined filter, select AND or OR and define the additional rules or groups.
- 4. If needed, select **Delete** to delete a rule or a group of rules.
- 5. Select Save.

#### Example

Have a look at the previous example with the database collection customobjects.customers. If you want the column only to display the customers that are blocked, you can use this filter:

info equal blocked

If you want to display the blocked customers from Belgium, you can use:

```
info equal blocked
AND
country equal BE
```



=

Note: Filter is not used with the \*Define own list\* collection.

**Note:** If you want to use the filter on a boolean, select (custom), enter the key value in the first field, followed by equal and true or false.

			_	
(custom) ᅌ	selected	equal	۵,	true

In this case, the filter will filter on all keys of which the value is true.

- Height: here you can define the height of the grid.
- **Topbar buttons**: here you can define buttons that appear at the top of the table. Select *s* to open the **Manage buttons** dialog. Each column represents a button with its icon and text above the table.
  - Select + or Add column to add a column.
  - Select 

     to drag the column to another position.
  - Select to remove the column.
  - Select Select to close the tab.
  - Icon: here you can select an icon you want to use for the button.
  - Label: here you can enter the name that appears on the button.
  - Action name: here you can define a name for an action. The action name appears in the section Actions in the component settings. For each action name there is a dedicated action. See Working with Actions on page 104 for more information.
  - Enabled when: here you can define when the button is available and the action can be triggered. Options:
    - Always: the button is always enabled.
    - No selection: the button is enabled if no table row is selected.
    - Only one selected: the button is enabled if only one table row is selected.
    - One or more selected: the button is enabled if one or more table rows is selected.
    - Multiple selected: the button is enabled if two or more table rows are selected.
    - **Conditional multiple**: the button is enabled if one or multiple rows are selected that match a conditional query. Select **\*** to open the tab to create the rule(s) for the conditional query.
    - Conditional only one: the button is enabled if one row is selected that matches a conditional query. Select
       to open the tab to create the rule(s) for the conditional query.
  - Tooltip: here you can enter a tooltip that appears when hovering over the button.

Select Save to save the configuration.

- Select: here you can define the selecting options.
  - **unselecting**: if you select this checkbox, you can deselect a selected item in the grid by reselecting it.
  - multiple: if you select this checkbox, you can select multiple items in he grid.
- Item Type: here you can select the item type. Options:
  - Image: if you select this option, the items in your grid consist of images.
  - Custom Page Components: if you select this option, the items in your grid consist of custom page components.
  - Item URL: here you can select the custom page component or define the URL to it. You can use a relative path.
- Item Width: here you can define the maximum item width.
- Item Height: here you can define the maximum item height.
- **Refreshing delay**: here you can select the delay after which the page is automatically refreshed. Select **No** refreshing if you don't want automatic refreshing of the page.

# Actions

With Actions you can manage actions that occur after a trigger. The following triggers are available:

• Row click: this trigger will start an action if you click the row.

- Double Row click:: this trigger will start an action if you double click the row.
- **action: Image click:** this trigger will start an action if you click an image in the table row. This trigger is only available if an image column is defined in the table **Columns**:

Manage columns		+ Add c
÷ •+	÷ 0+	
Туре	Туре	
text 🛟	image 🛟	
Title	Title	
Customer name	Image	
Key (path)	Key (path)	
cusname	imagekey	
Sortable Sortable	Sortable	
Searchable	Searchable	
Width	Max width	
Auto 🗘		
type specific options	Max height	
		Cancel

**Topbar: actionName**: this trigger will start an action if you click a button in the table Topbar. This trigger is only available if the button is defined in the **Topbar** buttons:

Manage buttons	+ Add colu	ımn
Icon    Label    Continue   Action name*   Continue_button   Enabled when		
Always 🕈		
Tooltip		
	Cancel	Sav

See Working with Actions on page 104 for more information on Actions.

•

# Dashboard container

The Dashboard container component adds a visual block.



The component has the following configuration options:

- Width: here you can define the width of the container.
- Height: here you can define the height of the container.
- **Background**: here you can define the background color.
- Border: here you can define the size and the color of the container border.
- **Border Radius**: here you can define the border radius that rounds the corners of the container's outer border edges.
- Margin: here you can select the top, bottom, left and right margins of the container.
- Shadow: here you can select the container's dropshadow properties:
  - The vertical and horizontal offset.
  - The amount of blur effect.
  - The color.
- Title
  - Show title: if you select this checkbox, the container title is displayed.
- Title text: here you can enter the title text. Select the pencil to open Expression Builder.
- Title style: here you can define the title style. Options:
  - - none -: if you select this option, the title doesn't have a defined layout.
  - **Primary**: if you select this option, the layout of the title will correspond with Primary style that you have defined in the Styling. See Styling on page 113 for more information.
  - Secondary: if you select this option, the layout of the title will correspond with Secondary style that you have defined in the Styling. See Styling on page 113 for more information.
- Source location: here you can define the source location of the container content. Options:
  - file: if you select this option, the content is an externally referenced file.
  - **embedded**: if you select this option, the content is embedded in the container. Select **Edit** to create or to edit the embedded content.
- Url (only available if Source location is set to file: here you can select the CLOUDFLOW URL of the external file. Select Go to editor to edit the file.

- **Input type**: here you can select the input type that sends data to the container content. Options:
  - none: if you select this option, no input is selected.
  - default: if you select this option, the default input type is selected.
  - **workflow**: if you select this option, you can specify a workflow to input data in the container content. In this case, a workflow is started when the container content is displayed. The variables generated by the workable can be used in the container content.

#### Custom page component

The **Custom page component** component adds another custom page component that was created with PAGEBUILDER.



A typical example of a custom page component is a header that you want to use on several pages. In case you want to edit it, you only have to change it on one location.

It has the following configuration options:

- Url: here you can select the custom page component or define the URL to it. You can use a relative path.
- Go to editor: if you select the link, the selected custom page opens in Pagebuilder editor and you can edit the custom page.
- **Input type**: the variables input type. Options:
  - none: if you select this option, no variables are passed to the URL.
  - default: if you select this option, the id parameters are passed to the URL
  - workflow: if you select this option, a CLOUDFLOW workflow runs to return data as a result.
    - Workflow name: the name of the CLOUDFLOW workflow.
    - Workflow input name: the input name of the workflow.
    - Open Workflow: if you select this option, the workflow opens in the Workflow Editor.

# Translate a page

Pagebuilder pages can be translated in case they are part of a Mars app.

There are multiple possibilities to translate strings in a Pagebuilder page:

- The translation can be provided in the Pagebuilder page itself.
- The translation strings can be exported to an XLIFF file. This way, the XLIFF file can be translated in a dedicated translation editor and can be imported into the Pagebuilder page when finished.

To enable the translations for a Pagebuilder page, open the Web Page Settings and select the checkbox next to **Enable custom translations**.

See Web Page Settings on page 69 for more detailed information about how to handle translations in a Pagebuilder file.

# Working with Actions

In various components, you can define Actions.

Actions can be for example a page refresh, a dialogue that is opened, a workflow that is started.... Actions occur after a trigger. The available triggers depend on the component you want to create an action for.

#### **Create an Action**

To create an action that is started by a trigger, follow these steps:

1. In the component settings, select a trigger in the Actions section at the bottom:

Actio	ons
$\mathcal{O}\mathcal{O}\mathcal{O}$	Row click Double click row <b>Select</b>

# =

Note: The available triggers depend on the component.

The following window opens:

Form: Actions				
Triggers	Actions			
∲ Submitted	Action: Block window O Block the complete window for the user			
	Waiting text waiting for workflow			
۵				
	Action: Wait C Waiting for "Set HTTP Reply" node of worflow			
	Action: Unblock window S Unblock the window			
	Action: Redirect to Go to an other web page			
	Url* redirect_fail.html			
	Open in new tab			
	Variables input default			

- 2. In the list of Triggers (A), select a trigger for the action. This trigger will start the action.
- 3. In Actions (B), select + Add to add an action. If there is a list of actions, the actions will be executed in a top-to-bottom order. To move the position of an action, hover over the action dialog until ... appears at the top of the dialog. Drag the dialog to the desired position. The execution of an action starts when the execution of the previous action has been completed.
- 4. Select the Action from the drop-down list.
- 5. Select Save.

#### **Actions overview**

These are the available actions:

This is an overview of the available Actions. You can create multiple Actions in one, that are executed consecutive. You can for example block a window, start a workflow, refresh a table and then unblock the blocked window:

Table: Actions		
Triggers	A	Ictions
Bow click		Action: Block window Block the complete window for the user
Double click row		
🗄 topbar: remove	0	waiting text removing items
占 topbar: reset	0	Action: Start flow Start a "start from kiosk" or a "start from form"
		Workflow name* pagebuildertests
		Workflow input name* remove_user
		Extra input options Key (path) Value
		Text 💠
		Action: Refresh table S Refresh the table
		Action: Unblock window C Unblock the window
		+ Add
		1

# **Refresh page**

• With this Action the page will be refreshed when the selected trigger is activated.

# **Redirect to**

With this Action you will be redirected to another web page when the selected trigger is activated.

- Url: here you can define the URL of the web page where you will be directed to. You can use variables to build up the URL. Select the pencil to open Expression Builder.
- Open in new tab: if you select this checkbox, the page will be opened in a new tab.
- Variables input: here you can define the variables to pass as parameters to the redirected URL. See Working with Page Variables on page 108 for more information on variables.

#### Dialog

With this Action a dialog box is displayed when the selected trigger is activated.

- Title: here you can define the title to appear in the upper left corner of the dialog box.
- Top: here you can define the position of the dialog.
- Width: here you can define the width (in pixels) of the dialog box.
- Height: here you can define the height (in pixels) of the dialog box. The default is auto.
- **Modal**: if you select this checkbox, the user is forced to interact with the dialog before the user can go back to using the page.
- Url\*: here you can define the URL of the page to be used to display the contents of the dialog box.

**Note:** You can only use Pagebuilder components URLs.

• Variables input: here you can define variables to pass. See Working with Page Variables on page 108 for more information on variables.

# Start Flow

With this Action a CLOUDFLOW workflow runs when the selected trigger is activated.

- Workflow name\*: here you can define the name of the CLOUDFLOW workflow. You should only use workflows that start with a **Start From Web Request** node.
- Workflow input name\*: here you can define the input name of the workflow.
- Extra input options: here you can define extra input variables to start a workflow.
  - Key (path): here you can define the key (path) of the variable that you want to pass to the workflow. You can use dot notation.
  - data type: here you can define the data type to interpret the data.
  - Value: here you can define the value of the key variable that you want to pass to the workflow.

# Close me

With this Action the dialog or disclosure panel closes when applicable.

# **Block Window**

With this Action you can block the window and add a text while the window is blocked.

# **Unblock Window**

With this Action you can unblock a blocked window.

#### **Refresh table**

With this Action the table will be refreshed when the selected trigger is activated.

# **Open Folder**

With this Action a folder is opened on your desktop. Enter the folder URL in the field. Select the pencil to open Expression Builder.

**Note:** This action is only available in case **CLOUDFLOW Plug-in Suite** is activated. See **CLOUDFLOW Plug-in Suite** on page 419 for more information.

# **Reveal File**

With this Action a file is revealed on your desktop. Enter the file URL in the field. Select the pencil to open Expression Builder.

**Note:** This action is only available in case **CLOUDFLOW Plug-in Suite** is activated. See **CLOUDFLOW** Plug-in Suite on page 419 for more information.

#### **Open Proofscope**

With this Action the asset is opened in PROOFSCOPE when the selected trigger is activated.

- Variable name with url\*: here you can define the name of the variable that contains an encoded CLOUDFLOW URL. This is the key name of the row object where the URL of the asset is linked to.
- Open with Cloudflow topbar: if you select this option, the CLOUDFLOW topbar is displayed.

# **Open Asset Viewer**

With this Action the asset is opened in ASSETS when the selected trigger is activated.

• Variable name with url\*: here you can define the name of the variable that contains an encoded CLOUDFLOW URL. This is the key name of the row object where the URL of the asset is linked to.

**Note:** This action is only available in case of image clicks. You can add image clicks by defining an image column in **Table** and **Job table** components.

# **Download file**

=

With this Action the asset is downloaded when the selected trigger is activated.

• Url: here you can define the CLOUDFLOW URL of the asset you want to download.

# Working with Page Variables

Pagebuilder allows you to use Page Variables.

You can apply Page Variables in Pagebuilder components and in Pagebuilder actions.

#### Components

If the Page Variables are active, you can use them in various components such as Title bar, Title, Text panel, Image...

Select the pencil to open Expression Builder..

Title			
	Text	The title is title	æ

You can define a default value for page variables as well. This will be applied if the variable does not exist.

To work with Page Variables in your components, select the **VARIABLES** button in the title bar. This opens a testing dialogue where you can choose the type of connector to retrieve the sample data to display as you create your table components.

- When the Page Variables are active, the arrow in the icon points to the right:
- When the Page Variables are not active, the arrow in the icon points to the left: 🛄.

**Attention:** The Page Variables must only return one object.

#### Context to test

In the first tab you can set and change the context to test the outcome of the use of Page Variables. Select a connector, define the settings and if needed select set to change the key. Select **Apply context to test** to apply the test context to the page.
Connector:	Preview variables:						
Cloudflow Collections (nucleus)	<pre></pre>						
Collection:	name: "Customer B"						
Custom Object: customers	happy: true						
Key:	}						
5ca4c3ed1bb5c317d32ee841 2	1						
/portal.cgi/PP_FILE_STORE/pagebuildertests/test/new.html?connecto =nucleus&collection=customobjects.customers&id=5ca4c3ed1bb5c3 7d32ee841	pr 1						
	Concol	Apply contex					

- **Connector**: **Cloudflow Collections (nucleus)**: if you select this option, Pagebuilder connects to the main CLOUDFLOW mongoDB database.
  - Collection: here you can define the mongoDB collection to retrieve data from.
  - Key: here you can define the unique \_id of the key of the document to retrieve. Select 🖼 to randomly change the \_id. A red border indicates that the \_id does not exist, a green border indicates that the \_id does exist. In case of an existing \_id, the available variables of the record are displayed under **Preview variables**.
- **Connector**: **Dataconnector**: if you select this option, any DATALINK connections setup on your CLOUDFLOW is connected to.
  - Collection: here you can define the name of the DATALINK query you have set up on your CLOUDFLOW.
  - Key: here you can define the unique key of the DATALINK document.
- Connector: Workflow: if you select this option, a CLOUDFLOW workflow runs to return data as a result.
  - **Note:** When triggering a CLOUDFLOW workflow for data, take the following into account:
    - Use a Start From Web Request node as the starting point of the workflow.

- ₫ 3 3 Start From Script Set HTTP End Web Request Reply (GetApprovals) ? X Set HTTP Reply Status: 200 \$ Content type: Contents: 1 result Delay: 10 ¢ seconds \$ Close
- Return the data contents as an object using an HTTP Response node as part of your workflow:

- Workflow name: here you can define the name of the CLOUDFLOW workflow.
- Workflow input name: here you can define the input name of the workflow.
- Input: here you can define extra input variables to start a workflow.
  - **use cache**: if you select this option, the previous data in the jacket is used.
  - Key (path): here you can define the key (path) of the variable that you want to pass to the workflow. You can use dot notation.
  - data type: here you can define the data type to interpret the data.
  - **Value**: here you can define the value of the key variable that you want to pass to the workflow.
- **Connector**: **Start from DBIO**: if you select this option, the test data is collected from a workable that was created by a workflow that starts with the Start from DBIO node and sets a reply with the Set DBIO Reply.
  - **Collection**: here you can define the collection created by running a workflow that starts with the Start from DBIO node.
  - Key: here you can define the unique key of the database document.
- **Connector**: **\*Custom**\*: if you select this option, you can select your own custom object as Pagebuilder context variables to test.
  - Input: here you can define extra input variables.
    - Key (path): here you can define the key (path) of the variable that you want to pass. You can use dot notation.
    - data type: here you can define the data type to interpret the data.
    - Value: here you can define the value of the key variable that you want to pass.

- Connector: File: if you select this option, you can select a JSON file as Pagebuilder context variables to test.
  - URL: here you can define the URL of the JSON file.
  - ID Key Name: here you can define the name of the ID key in the JSON to retrieve.
  - Key: here you can define the unique \_id of the key of the JSON file to retrieve. Select 🖼 to randomly change the \_id.

## Example

```
{
    "name": "Brett Pitt",
    "gender": "male",
    "company": "XYLAR",
    "email": "brettpitt@xylar.com"
},
```

In this example:

- The ID Key Name would be name.
- The Key would be Brett Pitt.

## Context when missing output

In this tab you can define a context variable on a page if no context is defined, in which case no id is available. If you want to provide a default variable in this case, you can define a value in the field **Value**.

### Example

The page **mypage.html** has a button. When clicking on this button, you will be redirected to a page **details.html**:

Button: Actions		
Triggers	Actions	
🗄 Click	Action: Redirect to   Go to an other web page	
신 Double click	Url* details.html	
	Open in new tab	
	Variables input none	
		c
<u> </u>		

These are the missing context settings for a page called **details.html**:

Context to test / Context when missi	ng input			۲
Connector:		Preview variables		
* Custom *	\$	▼ { name: "defaultName"		
Input:		}		
Key (path)	Value			
name Text 🜲	defaultName			
			Cancel	Save init context

In this example, the value defaultName is used when clicking on the button since there is no variables input.

## Actions

To pass variables as part of an action, you can define what to pass in the Actions dialog box of the components.

Here you can choose the variables associated with certain actions:

Table: Actions*			
Triggers	Actions		
🗄 Row click	Action: Redirect to O Go to an other web p	page	
b Double click row		ah aw Annya vala	
🗄 topbar: tt	011	snowApprovais	
		Open in new tab	
	Variables input	workflow	
	Workflow name*	pagebuildertests	
	Workflow input name*	dialog	
	Extra input options	Key (path)	Text \$

#### • Redirect to: Variables input:

- none: if you select this option, no variables are passed to the redirected URL.
- **default**: if you select this option, the id parameters are passed to the redirected URL for the selected or clicked table row(s).
- workflow: if you select this option, a CLOUDFLOW workflow runs to return data as a result.
  - Workflow name: the name of the CLOUDFLOW workflow.
  - Workflow input name: the input name of the workflow.
  - Extra input options: extra input variables to start a workflow.
    - Key (path): here you can define the key (path) of the variable that you want to pass to the workflow. You can use dot notation.
    - **data type**: here you can define the data type to interpret the data.
    - Value: here you can define the value of the key variable that you want to pass to the workflow.
- Dialog: Variables input:
  - **default**: if you select this option, the id parameters are passed to the redirected URL for the selected or clicked table row(s).
  - workflow: if you select this option, a CLOUDFLOW workflow runs to return data as a result.
    - Workflow name: here you can define the name of the CLOUDFLOW workflow.
    - Workflow input name: here you can define the input name of the workflow.
    - Extra input options: here you can define extra input variables to start a workflow.
      - Key (path): here you can define the key (path) of the variable that you want to pass to the workflow. You can use dot notation.
      - **data type**: here you can define the data type to interpret the data.
      - Value: here you can define the value of the key variable that you want to pass to the workflow.
- Open Proofscope:
  - Variable name with url\*: here you can define the name of the variable that contains an encoded CLOUDFLOW URL. This is the key name of the row object where the URL of the asset is linked to.
  - **Note:** This action is only available in case of image clicks. You can add image clicks by defining an image column in **Table** and **Job table** components.
- Open Asset Viewer:
  - Variable name with url\*: here you can define the name of the variable that contains an encoded CLOUDFLOW URL. This is the key name of the row object where the URL of the asset is linked to.
  - **Note:** This action is only available in case of image clicks. You can add image clicks by defining an image column in **Table** and **Job table** components.

## Styling

You can create a general style for your HTML page and apply it.

The general style will not overwrite the style of a component.

## Apply an existing style

How to apply an existing style.

To apply an existing style to the HTML page, select a style from the STYLE drop-down list at the top of the page.

To display the applied style, select the checkbox **Apply the style on the editor** in the **Web Page Settings** (a). Leave the checkbox unselected to hide the applied style and display the default style. However, the selected style will be applied to your HTML page.

### Create a new style

How to create a new style.

To create a style, follow these steps:

- 1. Select **STYLE** at the top of the page.
- 2. Select ADD NEW.
- **3.** Enter a name in the field **Name**.
- 4. If needed, you can copy the settings from an existing style. Select the existing style to copy from in the drop-down list in **Copy from**.
- 5. Select Create.
- 6. Specify the settings of the style. An example of the applied settings is displayed in the **Preview** window on the right.

If the HTML page is part of a MARS App, the \*.css styling file is added to a subfolder \_styles in the MARS App.

## General

Here you can define the general style settings.

- Color: here you can define the font color.
- **Background** : here you can define the background color.
- Font Family: here you can define the font family.
- Font Size: here you can define the font size.

## Style class

Here you can define the **Primary** and **Secondary** style classes. To add custom classes, select **+**. You can apply the style classes in the **Row conditional styling** parameter in the **Table** and **Job table** components.

- Font Color: here you can define the color of the style class.
- **Background**: here you can define the background color of the style class.
- **Border** : here you can define the color of the text frame's border.
- Font Style: here you can define the font style of the style class.
- Cursor: here you can define the cursor style of the style class.
- Border radius: here you can define the corner rounding of the text frame.
- On mouse hover:
- Font Color: here you can define the color of the style class on mouse hover.
- Background: here you can define the background color of the style class on mouse hover.
- **Note:** Component styles
  - Individual components can be set to override the style class.
  - Custom page components use the style of the page they are on.

## Table: Main

Here you can define the style of the main table.

- Background header: here you can define the background color of the table header.
- Border header: here you can define the color of the header's border. Options: none, 1px , 2px .
- Background odd rows: here you can define the background color of the odd rows.
- **Background collapsible**: here you can define the background color of the collapsible rows.
- Show row actions: here you can activate row actions. Options: Always and On row hover.
- **Background bottom**: here you can define the background color of the bottom buttons.
- Font color header: here you can define the font color of the header. Options: 5% increments between -25% and +25%.
- Font size header: here you can define the font size of the header.

- Background even rows: here you can define the background color of the even rows.
- Font size rows: here you can define the font size for the rows. Options: 5% increments between -25% and +25%.
- Row height: here you can define the height of the rows.
- Disabled paging buttons: here you can define to show or to hide the paging buttons.
- Font color bottom: here you can define the font color of a table bottom bar.

#### Button: Main - Button: Second

Here you can define the style of the Main and Second buttons. These buttons are used in the **Button** component and as a Submit button in the **Form** component.

- Background: here you can define the background color of the button.
- Background on hover: here you can define the hover background color of the button.
- Font Color: here you can define the font color of the button.
- Border Color: here you can define the border color of the button.

## Use case of PAGEBUILDER

This is a use case of PAGEBUILDER.

#### Who is involved?

#### The brand owner

**Chip** works as a project manager at **Reynholm Industry Brands**, a brand owner that among others produces energy drinks. The company works together with various suppliers such as design, marketing and translation offices to have their packaging created and translated, and printing companies to have it printed.

Reynholm Industry Brands' marketing partner has just delivered a new packaging design for the summer cherry flavor of the most popular canned energy drink product, **Star Drink**. Before ordering the offset printing job at their long-term printing partner, Chip wants to order some printed proofs first.

### The printing company

**Dale** is head of production at **Wernham Hogg Offset**, an offset printing company. The company is one of the suppliers of Reynholm Industry Brands.

#### What do they do?

To place the order for the printed proofs, Chip logs in on the customized order portal of Wernham Hogg Offset with his own username and password. He lands on a personal and private page for that was set up by Wernham Hogg Offset for Reynholm Industry Brands. The page contains the Reynholm Industry Brands logo and name, a list of jobs submitted, and a button to submit a new job:

1:	Subn	nit Job Job ID →	Customer Name	Job Description	Run Quantity	Approval State		modification
								start filtering
	۲	0827	0827 Customer	Coffeecups	1000	APPROVED	۲	08/30/2018 4:47 PM
	۲	0828	0828 Customer	Cupcakes	500	REJECTED	۲	08/30/2018 4:5 PM
	۲	0829	0829 Customer	Macadamia	1200	APPROVED	۲	08/30/2018 4:53 PM
	۲	0830	0830 Customer	Nut Bars	5000	APPROVED	۲	08/30/2018 4:50 PM
	•	0901	0901 Customer	Textpops	2000	APPROVED	۲	08/30/2018 4:59 PM

On the page, he clicks the Submit Job button and fills in the form with the job details and uploads the file

## How is the customized order portal set up?

The customized order portal is in fact a set of customized HTML page for Reynholm Industry Brands, built by Wernham Hogg Offset. It is called **list\_jobs.html** and it is an asset on their CLOUDFLOW system. It consists of the following components:

- A Custom page component of the Reynholm Industry Brands logo and company name called top\_bar.html. This page is itself a Pagebuilder page.
- A Jobs Table component which displays various columns of data about each Job.
  - A Topbar button which triggers the Action **submit\_job**, which launches the submit\_job.html page.
  - A Disclosure Content which displays showFiles.html, a Generic Table which is the file Chip uploaded.
  - A View Click action displays ShowApprovals.html, which is the result of the **GetApprovals** input from the **PB Exercise\_2\_Get\_Approvals** workflow.

To make sure that Chip only has access to the pages and assets for Reynholm Industry Brands on the Wernham Hogg Offset's CLOUDFLOW system, Wernham Hogg Offset has created a scope named reynholm, with a filter **PP\_FILE\_STORE/(reynholm/reynholm\_jobs)/.\*** and a welcome page cloudflow://PP\_FILE\_STORE/ reynholm/list\_jobs.html on their CLOUDFLOW system. The scope is assigned to a user Chip, which is the username that Chip uses to log in. The Pagebuilder pages will be stored at PP\_FILE\_STORE/reynholm, and the files that Chip uploads will be stored at PP\_FILE\_STORE / reynholm jobs.

# Workflows

CLOUDFLOW Workspace executes workflows.

## Workflows

A workflow is a sequence of predefined steps, which are configured with a specific set of parameters, and are executed step after step.

Here is an example of a simple workflow, introducing the main concepts:



- A: a workflow consists of **nodes**. These are the building blocks of a workflow. See Building a workflow on page 117 for more information.
- **B**: nodes are connected through **node connections**.
- C: nodes have input and output connection points.
- D: nodes can be configured via the configuration icon. If you Select this, the parameter window opens.
- A *workable* is an item that travels through the workflow. It is created when a item is submitted to the workflow and contains information about the item(s).
- Several workables can be combined in one *jacket*.

## Sub Flows

A sub flow is a workflow that is part of another workflow. It can be built within an existing workflow or made up as an independent workflow that the main work flow then links to to integrate it. Sub flows are built in the same way as workflows. Their integration typically uses the nodes:

- 1. Start Sub Flow on page 152
- 2. Call Sub Flow on page 143
- **3.** Return From Sub Flow on page 147

Sub flows are a common way to customize default work flows to individual needs.

## Customization with sub flows

Sub flows can be applied to customize workflows though this function is for advanced users. See Sub Flow Customisation on page 131.

## **Building a workflow**

Workflows are built with Nodes.

This topic describes routine processes of workflows building. More advanced functions can be found in Advanced Workflow Building on page 131.

Nodes are the bricks of a workflow. They start an activity in the workflow, depending on the previous nodes or activities. They are graphically represented by a block.

To build a workflow, you need to perform several steps:

#### 1. Drag the node to the workflow area

- 1. Select FLOWS > NODES. To search for a specific node, you can use the quick search window on top of the node list.
- 2. Drag the node from the nodes list to the workflow area on the right.
- Note: You can modify node label in the workflow area by double-clicking it. When you press the ALT button on your keyboard and hover over the node, it will show its original label. This is useful in case of workflows with a lot of renamed node labels.

You can delete a node from the workflow area by selecting it and pressing the Delete button on your keyboard.

You can copy-paste one or more nodes.

#### 2. Configure the node

To configure a node, select the configuration icon on the bottom right of the node in the workflow area. This opens the configuration panel. Select **Close** to close the node without saving the modifications. Select **Save** to close and save the modifications.

The configuration panel is specific for each node and allows you to configure the processing of the node. See Nodes overview on page 133 for more detailed information on how to configure the node parameters.

#### 3. Connect the node

Nodes have connection points.

#### Example

The Copy File node has three connection points.

- One input connection point (A).
- Two output connection points:
  - Success, where the workflow will go to when the operation has succeeded (B)
  - Failure, where the workflow will go to when the operation has failed (C)



However, depending on the node, the connection points can differ.

To **connect** a node with another one, select an input or output connection point from the node and move towards the input or output connection point of the node you want to connect with. When reaching this connection point, a connection is established which is represented by a connection line between the two nodes. This way, a sequence of operations that the workflow needs to perform is created.

In most cases, the success output of a node is connected to the next. To handle errors, the failure output of a node is connected to another node that needs to handle something (for example cleaning up files that to prevent them from remaining on the disk).

The end result will be a connected graph of nodes, which represents the workflow:



To **disconnect** a node from another one, select the node connection and move it towards an unspecified location in the workflow area (this means, not a connection point).

To **modify** an existing connection, select the input or output connection point and move towards the input or output connection point of the node you want to connect with.

## **Workflow Editor**

With the Workflow Editor you can create, select and modify workflows.

To open it, select FLOWS > WORKFLOWS.

CLOUDFLOW											,	LL SCOPE	s V	HELLO ADMI	NISTRATOR	•
- 🧲 🔒 Create Job	2							+	ADD C	0 M M E ?	νт 🤌	<b>1</b> 00	is 🕨 K	IOSK Q		
WORKFLOWS NODES WORKABLES							2.12	-	C		G	D	(E		• <b>(E)</b> •	
(P Name											_	-			-	
🗈 1 test										Pro	lob in duction					
Add files to variable in Job				St	art From	Create Jo		C Hold In					ि End			
Approval Flow Approval					Kiosk		P	Kiosk					٥			
Approval Flow approval evy										1	lob in					
ApproveWithForms											0					
la Asset Actions Flow																
🖧 cerm					R											
A Check in - check out					•											
Cloudflow-Share-1																
🖏 Сору														Unhandled		
ℰ» Copy folder														Problem		
A Create Job																
d Create Job 25																
ď≌ Create Job copy																
n create simple job															H	
+ 10 8 0 0	LOGS VARIABLES	OUTPUT	RES-PARAMS	EXE-PARAM	5											Q

**Note:** In case the Workflow Editor is by default set to view only, you cannot edit any workflows. Select the lock icon on the left of the workflow name to unlock the Workflow Editor and edit the workflow or the nodes. Select the icon again to lock the workflow to prevent accidental user edits in the Workflow Editor. You can manage the default behavior of the Workflow Editor protection in **SETTINGS** > **SETTINGS** > **QUANTUM** > **EDITOR PROTECTION** 

It consists of the following parts:

- A: the workflow list, where all the workflows are listed. If you select a workflow, it is displayed in blue.
  - Select at the bottom to add a new workflow. A new entry will appear in the list, you can change the name of the workflow by double-clicking the name in the list, and entering a new one. See Building a workflow on page 117 for more detailed information on how to build workflows.
  - Select 🖸 to copy an existing workflow. If you copy a workflow, the hotfolders will be disabled in the copied workflow.
  - Select 💼 to remove an existing workflow. It is not possible to remove a workflow with running jobs.
  - Select to export an existing workflow.
  - Select <sup>1</sup> to import an existing workflow.

You can use the quick search on top of the workflow list to search for workflows:

- If the query matches a workflow name or a part of a name, the list is filtered by the matching workflows and the query is highlighted in yellow.
- If the query matches node name or a part of a name, the list is filtered by the workflows that contain nodes with the matching name and a magnifier icon appears.

#### Example

In this example, the query **Jobs** has the following result:

WORKFLOWS	NODES	WORKABLES
(P Job		
Create Job		Q
🚯 <mark>Job</mark> s		Q
Standalone F	RIP	Q
A 1test		٩

- The query matches two workflows because **Jobs** is present in the name of the workflows (**Create Job** and **Jobs**).
- The query matches four workflows because **Jobs** is present in one or more node names or types in the workflows.
- **B**: the workflow area, which is the graphical representation of the currently selected workflow. By manipulating this graphic, the workflow can be changed:
  - Extra workflow steps can be added.
  - A workflow step can be connected or disconnected from another to change the processing flow.
  - Each step can be configured with different parameters.
  - ..

See Building a workflow on page 117 for more information.

- C: Add Comment. With this button, you can add a comment to the selected workflow.
  - Select + to add a comment. A text box will appear.
  - If you hover over the text box, a pencil will appear.
  - Select the pencil to edit the comment.
  - Select the text box to move or resize the comment field.
  - Select the text box and press the **Delete** button to delete the comment.

- D: Logs. With this button, you can view the timeline of the selected workable. Each workable has its own timeline. If you select a different timeline, the corresponding workable is selected in the workflow.
- E: **KIOSK**. With this button you can access KIOSK and submit a job to the selected workflow. Follow these steps:
  - 1. Select the workflow you want to submit to from the workflow list on the left.
  - 2. Select the **KIOSK** button (**D**) on the top right in the workflow area.
  - 3. Select the Input.
  - 4. If needed, provide extra parameters for the workflow. See Form Builder on page 260 for more information on how to predefine these parameters.
  - 5. Browse to the file(s) or select them from a File store. See Enable File store browsing in the Start from KIOSK node for more information on file uploading.
  - 6. Select Submit.

You can also use this for workflows that start with the **Start From Web Request** node. In that case, you can add extra input options to the workflow:

- Key (path): here you can define the key (path) of the variable that you want to pass to the workflow. You can use dot notation.
- data type: here you can define the data type to interpret the data.
- Value: here you can define the value of the key variable that you want to pass to the workflow.

- F: the Quick search field where you can enter a search string.
  - An orange font in the node name indicates that the query is found in the node name .



For example Job in Production

• An orange border around the node indicates that the query is found in the node **type**.

For example Create Job.

	+ ADD COMMENT	►кіозк Q	Create Job
Start From	Hold In	Job in	End
Kiosk	Kiosk	oduction	©

The node type depends on the language of the CLOUDFLOW User Interface. For example, in a French User Interface, the **End** node type is **Fin** and the query **End** will not have any results.

• An orange colored node indicates that the query is found in the **field values** of the node.

For example In Review (because the State field value is IN REVIEW).



Some queries have combines results. For example, the query Job in the previous example will result in this:



• G: heatmap: the heatmap indicates the processing time of the workable in each node of the workflow. If you draw a heatmap, the nodes (except for some nodes such as for example the Start, End, Hold In Kiosk, Wait For Approval ... nodes and nodes that take more than 24 hours for the workable to be processed) will appear in a color that varies from dark green to dark red. The node that is colored in dark green is the node with the fastest processing time, while the node that is colored in dark red is the node with the slowest processing time. The other nodes are painted in a color between dark green and dark red, depending on the processing time. In case of differences, a legend is shown in the topbar:



- **Note:** The processing time of the nodes is relative to the run time of the workable in the workflow and does not depend on the system. For example, this means that a red node does not necessarily mean that it is problematic, it just means that it is a the processing was slow compared to the processing times of the other nodes.
- H: Zoom: with these buttons you can zoom in and zoom out in the Workflow Editor.
- I: Debug area: see Debug area on page 128 for more information.
- Note: Some workflows that are based on a template have a checkbox Auto update (for example
   Standalone RIP). If you select the checkbox, the workflow will be reset to the default system setting when

CLOUDFLOW is updated. This means that all changes to the template based workflows are lost with every CLOUDFLOW update. If you want to keep the changes, make sure to leave the checkbox **Auto update** unselected.

=

**Note:** When saving a workflow, a small rotating icon is visible on left of the zoom button, until the saving is done.

#### **Workflow Editor shortcuts**

	ŵ.	<i>*</i>
Copy selected nodes and comment fields	Cmd + C	Ctrl + C
Cut selected nodes and comment fields	Cmd + X	Ctrl + X
Paste selected nodes and comment fields	Cmd + V	Ctrl + V
Add to selection	Shift + click	Shift + click
Show node more details (for example the names of the node and the connection points)	Alt + mouse hover	Alt + mouse hover
Add nodes to selection if they are inside a comment field and the comment field is selected	Alt + comment click	Alt + comment click
Delete selected items	<ul><li>Cmd + Backspace</li><li>Backspace</li><li>Del</li></ul>	<ul><li>Ctrl + Backspace</li><li>Backspace</li><li>Del</li></ul>
Save	Cmd + S	Ctrl + S
Return to previous situation	Cmd + Z	Ctrl + Z
Undo	Ctrl + Shift + Z	Ctrl + Shift + Z

Note: Restrictions for 윋:

- Use an international key board.
- Use the numeric keypad.

## WORKABLES tab

F

In the WORKABLES tab, you can see an overview of the jackets and workables.

When you have submitted a job to a workflow, a jacket and a workable are created.

## Jackets list

The **WORKABLES** tab provides a list of the jackets in the selected workflow.

On the bottom, you can activate various filters on the list:

- *I* hides the old jackets. These are the jackets that existed before selecting the icon. If you submit a new job while the filter is active, the new jacket is displayed in the list.
- **x** shows all jackets that contain a workable with an error state.
- > shows all jackets that contain running workables.

Note: You can activate various filters at the same time.

If you select a jacket, you can delete (\*) and download (\*) it.

## Workables

=

If you select a jacket, it opens and it shows an overview of the workables in the jacket:

\_\_\_\_\_

WORKFLOWS	NODES	WORKABLES			
▼ 20160513091640_89473					
modification: 12/7/2017 12:16:21					
20160513091640_89473					
state: done					
THE 20160513091640 89473					
20160513091640_89	473 - admin@	hybrids			
wor	kflow: Approva	I Flow Approval			
	file:				
20160513091632	2_86459				
20160513091622	2_51964				
20160513091613	3_42740				
123456	▶ 123456				
12345					
12345-004					
12345-003					
12345-002					
12345-001					
12345-004					
12345-001					
0801014B_75ML	_EDT				
200131-001P001	_Freigabe				
200131-001P001					
186735-001P001	_Freigabe	1			
186735-001P001					
23456_S&R					
23456					
12345_S&R					
12345					
▶ 1Color					
2					

Workables can have two different states:

• running: this means that the workable is still running.



In this case, you can perform actions on the workable:

- II: if you select this icon, the workable will be paused at its current the location. The icon will change into >.
- **•**: if you select this icon, the workable will start running again.
- M: if you select this icon, the currently paused workable will move one workable.
- **x**: if you select this icon, the workable will be deleted.
- done: this means that the workable has reached its end point. You cannot perform any action on the workable.



#### Representation in the workflow area

When you select a workable, it becomes visible in the workflow area. The blue line represents the workable path:

CLOUDFLOW		ALL SCOPES Y HELLO ADMINISTRATOR
- R APPROVAL FLOW /	NPPROVAL	🕂 ADD COMMENT 🕨 KIOSK
WORKFLOWS         NODES         WORKABLES           ▼ 20160513091640_89473         x ▲           modification:         127/2017 121621           ▼ 20160513091640_89473         x	Blaut From Propare Oct File Professoor Problem	Report File Charged
water Approval Flow Approval     water Approval Flow Approval     water Approval Flow Approval     water Approval Flow Approval     water App	Stort Approval 1	Route Aher Approval 1
20160513091632_86459     20160513091632_874     20160513091632_51964     20160513091613_42740     123456     409.45	Constato PURE 1 URE 1	Rejected
+ 12345-004 + 12345-003 + 12345-002 + 12345-001 + 12345-001 + 12345-004	Report 5- Mail problem	S Run Falure
12345-001     08010148_75ML_EDT     200131-001P001_Freigabe     200131-001P001     186735-001P001_Freigabe     186735-001P001	C Generic Palure Palure Approval Poternal	
> 23456_S&R > 23456 > 12345_S&R > 12345 > 12345	Assessme to the second se	
8	.OGS VARIABLES OUTPUT RES-PARAMS EXE-PARAMS	+ -

If the workable is still running, a green dot indicates the current position of the workable:



When the workflow is stopped, for example in case of an error, you can move this green dot to another node. If you select  $\blacktriangleright$ , the workflow is restarted from this location without having to reload the job.

### Debug area

At the bottom, extra options are available with information about the selected workable. These buttons are used mainly for debugging reasons. Select <sup>(2)</sup> to copy the content of the panel as text to the clipboard.

To view the information, select a workable and one of the available options:

- LOGS: here you can see log information of the selected node. If the log contains encoded CLOUDFLOW URLs, you can hover to see the decoded forms.
- VARIABLES: here you can see an overview of the variables that have been generated by the workflow at the moment of the current position of the workable. Variables that not yet have been created are not listed.
- **OUTPUT**: here you can see an overview of the output data generated by the selected node. There are three types of output:
  - References: the datatype is an object with key files and the value is an array of file URLs.
  - Variables: the datatype is an object with a key and a value.
  - Data: the datatype is an array containing objects with two keys: type and URL.

#### Example

You have the following example workflow with the following parameters:

Start From Set name XIML to Set sumame	Contraction of the second seco
Set name (Set Variable) ? Variable name Value iname MC C	XML to JSON     ?       XML Data Type:     Auto detect       XML Data:     ?       YML Template Type:     XML Data       XML Template Type:     XML Data       XML Template Type:     ?       YMM version="1.0" encoding="UTF-8"?> captrices celoremic lobid="12345">
Close Save	<pre><dbs: dx="" s<="" sci="" th=""></dbs:></pre>
sumame Hammer <table-cell></table-cell>	Incomplete Template:     Allow Incomplete Template:     Strip Whitespace:     Output:     Output:     Output:     Overwrite existing file
Close Save	Close Save

You run the following XML file through the workflow:

```
<id>l2345</id>
</id>
```

• If you select the **Start From Kiosk** node, you see an example of a **reference** output type:

```
[
{
pass: 1,
references: {
files: [
"cloudflow://PP_FILE_STORE/12345.xml"
]
}
]
```

• If you select the Set name node, you see an example of a variables output type:

```
[
{
pass: 1,
variables: {
name: "MC"
}
]
```

• If you select the XML to JSON node, you see an example of a data output type:

```
[
{
pass: 1,
data: [
{
   type: "com.nixps.quantum.data_provider.0",
   url: "cfqworkdata:///4c7aa1d5-545e-4d47-92c9-309e49cdb3f4/"
}
]
}
```

The resolved output is calculated but it is not stored in the workable. It is accessible for **File from node** and **Data in Container from previous**. You can also define the node name in stead of previous, but it will use what is available as resolved output from that node. This means that references and data are kept when a node does not generate either files or data. In the example above, the output and resolved output from the **Start from Kiosk** node will always be the same, as this node is the first one. If you select the **Set name** node, the resolved output looks like this:

[

```
{
references: {
files: [
"cloudflow://PP_FILE_STORE/12345.xml"
]
}
,
pass: 1,
variables: {
name: "MC"
}
]
```

This node didn't generate any references, however they are still available, so resolved means what is available at that time. It will also combine the output with the output which the node created, this is the variable name. This is rather redundant as variables are always direct accessible. The **XML to JSON** node will use that reference, as the XML Data is set to use the full file URL from the previous node. The **XML to JSON** node only generates data, but the resolved output shows this:

```
[
{
references: {
files: [
"cloudflow://PP_FILE_STORE/12345.xml"
]
}
,
data: [
{
type: "com.nixps.quantum.data_provider.0",
url: "cfqworkdata:///4c7aald5-545e-4d47-92c9-309e49cdb3f4/"
}
pass: 1
}
```

The reference doesn't change as no new reference is generated. However, the resolved output and the content of the data in the container are shown:

```
ſ
{
url: "cfqworkdata:///4c7aa1d5-545e-4d47-92c9-309e49cdb3f4/",
data: {
entries: {
element: {
jobid: 12345,
id: "12345",
company: "Hybrid Software",
address1: "Guldensporenpark 18 block B",
address2: "--",
zip: "9820",
city: "Merelbeke",
country: "Belgium",
phone: "+32 9 329 57 53",
email: "info-eur@hybridsoftware.com",
description: "PACKZ Green Tea Lemon",
asm: "Jack Sparrow"
```

```
tablename: "main"
}
]
```

To summarize: resolved data can be the above three types of output, and will not change if a node doesn't change them, except for variables. Data is the content of the data in container.

- **RES-PARAMETERS**: here you can see the RES parameters of the selected node.
- EXE-PARAMETERS: here you can see the EXE parameters of the selected node.
- ADD BREAK: with this button you can add a break to a selected node to make the workable stop at that specific node. This can be useful for testing reasons. Select REMOVE BREAK to remove the break.

## Advanced Workflow Building

This section explain functions that exceeds routine use and that are for advanced users.

## **Sub Flow Customisation**

Sub flows can be applied to customise workflows though this function is for advanced users.

To refresh what sub flows are, please read the topic Sub flows.

Sub flows can customize a workflow in two ways:

- 1. A separate sub flow can be created to add operations to an existing workflow.
- 2. In an existing workflow certain operations can be regrouped to form a sub flow that is "quarantined".

This second process has two definite advantages:

**a.** A "quarantined" sub flow can be skipped, replaced by another flow or can itself be integrated intó another flow.

This is useful because not all users require all workflows to run fully. Some, indeed, have individual needs that they want to see a default workflow implement. The best method then is to group the unnecessary default operations into a sub flow that is then skipped. This process is called **hooking** and the sub flows are said to be pegged up by **hooks** 

**b.** Quarantined/Hooked sub flows can be replaced by others without their host workflow being considered modified and so barred from software updates.

Hooked sub flows are managed in the Workflow Hooks tab. See Workflow Hooks on page 131.

#### **Workflow Hooks**

In this tab you can redefine sub flows by using Hooks. The function is for advanced users.

### **Hooks: Definition and Conditions**

Workflow Hooks are the delimiters of sub flows. By redefining them you can make a Call Sub Flow node in a work flow call another sub flow than was originally intended to.

The Workflow Hooks tab displays a list of hooked sub flows that match the following conditions:

- The sub flow must have been configured with a Hook Identifier.
- The sub flow must be called by another work flow.
- The sub flow must have run at least once.

### **Hooks: Redefining**

For the process to be illustrated it requires the presence of 2 ready made flows, 1 work flow and one sub flow.

Build them following the topics Building a workflow on page 117, Start Sub Flow on page 152, Call Sub Flow on page 143 and the instructions below:

=	උmy_workflow0i දියිදියියි. අන්තර සිටින් සිට					
∩ ₽ ₽	Home Notifications Assets	Workflows         Nodes         Worka           Q_my_         my_workflow01         Image: Compare the second sec	Start From Klosk	Call Sub Flow	End Input Input Input Name Original	0 ×
	Approval Tasks Jobs RIP Klosk Flows VDP Tectonics Mars Deabboard Settings Manual	my_test_hook     my_test_hook     My_touvertest	Store From Sold     Sing     Sing	It sab phere I and the second	Options Under Control of Problem Cation Rev	
Q	Administrator		Number of Hits (Note Accessed Miner Syste Workshile name Unit To Jackst Burstmell Unit To Jackst Burstmell Scope From User		Venders For Sub Proc. (Al Venders from vorkalie v Recurs Vander Name (ed), fore vende Close (ed)	, Q Q

- 1. In the Workflow Editor, **build a basic workflow** as described in Building a workflow on page 117 and configure it as the example image above *my\_workflow01*. Start with a **Start from Kiosk** node and end it in an **End** node.
- 2. Drag in a Call Sub Flow node. This will call the sub flow built in step 3 below.
- 3. To integrate a sub flow, drag in the Start Sub Flow and Return From Sub Flow nodes.
- 4. Configure the nodes as per example.

Make sure to name the sub flow consistently in all node configuration dialog boxes.

- 5. Click open Kiosk and Submit to run the workflow.
- 6. In the Workflow Editor, **build a sub flow**, like in the example below *my\_test\_hook*. This will serve to substitute the first sub flow.

=	A my_test_hoo	k					ጜ   ጜ
$\square$	Home	Workflows Nodes	Workables				<u> </u>
Û	Notifications	Q my		3		Return	
	Assets	my_test_hook		Start Sub Flow		Flow	d Problem
0	Approval	ny_workflow01			Start Sub Flow	0	
<b>6</b>	Tasks	A MyCLOUDFLOW			Name: my_test_hook		
=	Jobs	VDP-CORE	0		Input parameters:		
-	RIP	n dfe_core-timer	0		Check Input parameters		
	Kiosk	n eyec-Proofiler Driver	Q		Check input parameters		
No.	Flows	cf_whats_new-21-02	0				
	Testonics						
-	Mare						
	Settings					Close Save	
Ô	Manual						
-							

7. Click the Manage Hooks button it to open the Workflow Hooks tab.

The hook is listed in the **Workflow Hooks** tab.

	Workflow Hooks				Ę	5 0
$\triangle$	Home	Identifier	Workflow	Sub flow	Save Workflow Hook	×
Û	Notifications				Montifier 177	
	Assets	123	my_workflow01	my_test_hook		
0	Approval				Workflow my_workflow0	
勴	Tasks				Sub flow my_test_hook	
:=					Cancel	Save
	RIP					
	Kiosk					
Q_	Flows					
 ≪	VDP					
-	Tectonics					
소	~~~	hannen	man man	man market and the second s	La companya and a com	$\sim$

Should the list be long and you cannot find the hook, you can use the quick search on top of the **Workflow Hooks** list to search them. If the query matches an entry name or a part of a name, the list is filtered by the matching hits and the hit is highlighted in yellow.

8. Double click the hook line.

The hook's details pane is displayed Save Workflow Hook.

- 9. Enter the name of the second sub flow so as to point the main workflow to it.
- 10. Click Save.

When you now run the first workflow, you will see that the sub flow is skipped and the its nodes display hook marks.

## Nodes overview

The following chapters will give you an overview of all nodes and their parameters.

Nodes are divided into several categories. The availability of the nodes depends on your license.

## Input

Input nodes start a workflow.

#### **HTTP Service**

With this node you can trigger the running of a JavaScript as response to a web REST call to CLOUDFLOW. One of the use cases is to allow a REST call to start a workflow.

It has the following configuration options:

- Include Scripts: here you can define the list of script files that you want to include. Select + to add a script file.
- Name: here you can specify the name of the configured service.

#### Calling an HTTP service

An HTTP service can be called in a workflow by a **Call REST** node. The URL needs to consist of the following parameters:

```
http://<server>:<port>/
http_service=<name_of_the_service>&whitepaper=<name_of_the_workflow>
```

#### Example

CLOUDFLOW	ALL SCOPE	S ~ HELLO ADMINISTRATOR	
HTTP SERVICE	+ ADD COMMENT	► KIOSK Q	
WORKFLOWS       WORKABLES       ? *         WORKFLOWS       WORKABLES       Performance         A treat	Unhandled Problem		
New Flow 22      LOGS VARIABLES OUTPUT RES-PARAMS EXE-PARAMS ADD BREAK			Q

Important parameters in this example:

- A: HTTPservice is the name of the service.
- **B**: **http Service** is the name of the workflow.
- C: this means that the URL to invoke the HTTP service is http://127.0.0.1:9090/portal.cgi? http\_service=HTTPservice&whitepaper=http%20Service.

## Setting the output of an HTTP Service

You can set an output of the HTTP service in the script of the HTTP Service node with the parameter setOutput.

## Example

НТ	HTTP Service ?					
	Include scripts:					
	Nar	ne: HTTPservice				
12	1 2 3 4 5 6	<pre>auth.login('admin','admin'); console.log('test 1 successfull'); var l_json={'variable1':'value1','variable2':['value2','value3','value4']} file write icen file(h icen 'cloudflow (/DD ELE 5 57005 (test));test)</pre>				
2	8	Tite.write_json_tite(i_json, cloudtlow://PP_FILE_STORE/testhttpservice/testi.json ,{ overwrite	e :true	3)		
i	9	setOutput("ResultOK")				
			Close	Sav	ve	

In this example the output of the HTTP Service will be a variable **ResultOK**.

## Setting the security of an HTTP Service

You can manage the authentication of an HTTP Service in the **HTTP Service** node and pass the credentials via the **Call REST** node.

#### Example

The user credentials are entered in the **Request Data** parameter field in the **Call REST** node:

-						
C	all REST 3 (Call REST)		?			
	HTTP Method: POST $\diamond$ Content Type: application/json $\diamond$ URL:					
	http://127.0.0.1:9090/portal.cgi?http_service= service &whitepaper=httpSe	ervice	1			
	Name Value	Θ				
	Request Data:					
	{"login":"admin","passw":"admin","other_data":{"var1":"val1","var2":"val2"}}		· ·			
L	Upload files: None  Field name for uploaded file:  File to upload: Verify SSL Certificate:  SSL Version: Automatic  Put result in variable: rest_result Put response headers in variable:					
			Close Sav			

In this example the credentials are the following:

- Log in: admin.
- Password: admin.

The authentication is managed in the HTTP Service node:

HTTP Service - 3 (HTTP Service) ? X							
Inclue	Include scripts:						
Nan	Name: test3						
1 2 3	<pre>addMessageToLog("info", getInput()); //getInput() returns data as string</pre>						
4 5 6	<pre>var l_values = JSON.parse(getInput()); console.log(l_values);</pre>						
7 8 9	<pre>// check if call is send with the right credentials var l_session=auth.login(l_values.login,l_values.passw);</pre>						
10 11 12 -	<pre>console.log(l_session); if (l_session.user_hash===undefined){</pre>						
13 14 - 15	<pre>addMessageToLog("error","Unknown user tries to connect with HttpService test3")} else{     //looks like we have a valid user</pre>						
16 17 18	<pre>addMessageToLog("info","Valid User tries to connect with HttpService test3"); var l_start=hub.start_from_whitepaper_with_files_and_variables('httpService', 'form'.</pre>						
19 20 21	[], {'resto':'Chez Janou','meal':'dinner','names':'Jeff,Jos,Janine,Jack,John'}); var l_workable_id=l_start.workable_id:						
i 22 23 i 24	<pre>console.log(l_workable_id+'') setOutput('Workable_id='+l_workable_id)</pre>						
25 26 27	}						
	Close Save	Ð					

In this example, two scenario's can occur:

- A user that is logged in with username **admin** and password **admin** submits a job to this workflow. In this case the credentials are valid and a CLOUFLOW session is created which triggers another workflow.
- A user that is logged in with a username or password that is different from **admin** / **admin** submits a job to this workflow. In this case the credentials are not valid and the following log is created: Unknown user tries to connect with HttpService test3.

## Converting script functions results to JSON format

In a script you can use functions. See Functions in the Script node on page 464 for more information.

The function **getInput()** returns a string. To convert this string to JSON, you can use the function **JSON.parse(getInput())**.

Example

HTTP Service - 3 (HTTP Service) ? X						
Include scripts:						
Name: test3						
<pre>1 addMessageToLog("info", getInput()); 2 3 //getInput() returns data as string 4 var l_values = JSON.parse(getInput());</pre>						
<pre>5 console.log(1_values); 6</pre>						
7 // check if call is send with the right c	redentials					
<pre>9 var l_session=auth.login(l_values.login,l 10 console.log(l_session); 11</pre>	_values.passw);					
<pre>12 if (l_session.user_hash===undefined){ 13 addMessageToLog("error", "Unknown user 14 elsef</pre>	tries to connect with HttpService test3")}					
15 //looks like we have a valid user 16 addMessageToLog("info","Valid User tr	ies to connect with HttpService test3");					
<pre>17 var l_start=hub.start_from_whitepaper 18 'form', 10 53</pre>	_with_files_and_variables('httpService',					
<pre>19 LJ, 20 {'resto':'Chez Janou','meal':'dinner' 21 var l_workable_id=l_start.workable_id</pre>	,'names':'Jeff,Jos,Janine,Jack,John'}); ;					
<pre>i 22 console.log(l_workable_id+'') 23</pre>						
<pre>i 24 setOutput('Workable_id='+l_workable_i 25 26 26 27 27 28 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	1)					
20 } 27						
	Close Save					

In this example, the info in the log in case of invalid user credentials is returned a string and converted to JSON.

## Start Cleanup

With this node you can start a cleanup.

If you add this node to your workflow, a new workable that starts a cleanup will be created when you select in **KIOSK** to remove a jacket.

**Important:** The workable will be created in the same jacket. Therefore, the jacket will not be deleted unless the new flow path ends with the **End** node and the checkbox **Delete Jacket** is selected.

The node has the following configuration options:

• **Priority**: in this drop-down list you can specify the priority to assign to the created workable.

## Start From DBIO

With this node you can start a flow and create a workable to collect and later return data to the **database\_proxy** module in the CLOUDFLOW API.

### Some words about CRUD

CRUD refers to the major functions that are implemented in relational database applications. It is short for:

- Create: a function that creates a new document in the database.
- Read (Get): a function that reads (gets) a document by id in the database by id.
- Update: a function that updates a document, where the document data contains enough information to find the document. No document will be updated if the document is not found.
- Delete: a function that removes a document by id.

### **CRUD** functions in the CLOUDFLOW API

The CLOUDFLOW API contains the **database\_proxy** module. This module contains functions to access data that is generated by workflows starting with the **Start From DBIO** node and that acts like a database. The module is used to implement CRUD functionality, compatible with our other CRUD functions, but that is fully customizable by using a workflow. See API on page 389 for more information on the CLOUDFLOW API. See the database\_proxy module in the API reference on http://<server>:cyport>/?api.

If you call one of this functions, a workflow is triggered.

### Example

You use the API call api\_async.database\_proxy.list(). In that case, the creation of a workable is triggered in the **Start From DBIO** node. In this workable you can collect the required JSON data that you need to return in the **Set DBIO** Reply node. The result of the above API call you will be the returned JSON.

With this method, you can implement CRUD functionality that can be used in your JavaScript files or in Pagebuilder and work with the data as if they come directly from a database.

## The node

It has the following configuration options:

- Name: here you can specify the input name to use in the API calls to select this starting point.
- Allow Access to Scopes: here you can specify which scopes are allowed to use this input. Options:
  - If you leave this field **blank**, users with the same scope as the Whitepaper are allowed to use this input.
  - If you enter \*, all users are allowed to use this input.
  - If you enter **a comma separated list of scopes**, users that are in one of the specified scopes are allowed to use this input.
- Allow Access to Users with Attributes: here you can define which attributes users need to be allowed to use this input. Options:
  - If you leave the parameter field black, all workflows will be accessible for all users in KIOSK.
  - If you define a comma separated list of attributes, all users who have one or more of the specified attributes are allowed to use this workflow in KIOSK.
  - **Note:** Only if these users also have the correct scope as configured in Allow Access to Scopes, they will be able to access the workflows in KIOSK.
- Enable for guest users: if you select this checkbox, guest users can use this node to create workables.
- Workable name: here you can specify the name to assign to the created *workable*. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- Link To Jacket (by name): here you can specify a *jacket* name that will be searched.
  - If found, the new workable will be linked to the jacket.
  - If not found, a new jacket will be created, and the specified name will be assigned to it.

Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.

- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.
- Scope From User: if you select this checkbox, the scope for the workable is taken from the user that is creating the workables. If you do not select it, the scope from the selected Whitepaper will be used.

Each output of the node corresponds to one of the CRUD functions:

- Get: if you select this output, a document in the database is read by id.
- Create: if you select this output, a new document will be created in the database.
- Update: if you select this output, a document in the database is updated in case the document data contains enough information to find the document. If the document is not found, it will not be updated.
- Delete: if you select this output, a document in the database is deleted by id.

Next to the CRUD functions, three extra outputs are available to execute extra functions:

- List: if you select this output, all documents that are specified by the supplied query are collected.
- Count: if you select this output, all documents that are specified by the supplied query are counted.
- **Get Columns**: if you select this output, the columns in the documents are returned. The return value of this function must be an object containing one key **columns** and a value with an array structure. Each element of the array must be an object. Each object must have a parameter name with an nonempty string containing the name of the column and a parameter **data\_type** containing the JSON type representation of the internal value of the column. Supported values are **number**, **string**, **boolean**.

```
"columns" : [{
  "name": <name of the column, JSON path to the value>,
  "data_type": <"number" | "string" | "boolean" | (not yet supported but
  allowed "date", "date_iso")>
}]
```

```
=
```

**Note:** It is recommend that the **name** and the **data\_type** correspond to the returned JSON structure, because some features will directly handle the returned JSON and do not pass via the **Start from DBIO** node.

#### Start From Web Request

With this node you can trigger the creation of a workable and/or jacket by submitting a web request.

It has the following configuration options:

- Name: here you can specify the name that has to be used by web pages to reach this input node.
- Allow Access to Scopes: here you can specify which scopes are allowed to use this input. Options:
  - If you leave this field **blank**, users with the same scope as the Whitepaper are allowed to use this input.
  - If you enter \*, all users are allowed to use this input.
  - If you enter a comma separated list of scopes, users that are in one of the specified scopes are allowed to use this input.
- Enable for guest users: if you select this checkbox, guest users can use this node to create workables.
- Workable name: here you can specify the name to assign to the created *workable*. You can leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- Link To Jacket (by name): here you can specify a *jacket* with the name that is specified here will be searched. If found, the new workable will be linked to the jacket. If not found, a new jacket will be created, and the specified name will be assigned to it. Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.
- Link To Jacket (by ID): here you can specify the parameter that will be used to search for an existing jacket that uses the specified ID. This allows you to link the new workable to an existing jacket. You can also look up a jacket by name by using the Link To Jacket (by name) parameter. This parameter takes priority over the Link To Jacket (by name) parameter while searching for a jacket. Select the pencil to open Expression Builder.
- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.
- Scope From User: if you select this checkbox, the scope for the workable is taken from the user that is creating the workables. If you do not select it, the scope from the selected Whitepaper will be used.

#### Start From Hot Folder

With this node, a *workable* will be created when a new file appears or when an existing file is updated in a specified folder.

It has the following configuration options:

- Hot folder: here you can specify the folder that will be monitored for new or updated files.
- Settling time: here you can specify the minimum number of seconds that should elapse before the actual processing on the file begins. This is necessary when large files are copied or when copying is slow, and processing should wait until the file is completely copied.

- Enabled: if you select this checkbox, the hotfolder will be monitored. If you do not select it, the hotfolder will not be monitored.
- Enable "Deleted File" handler: if you select this checkbox, a workable is created when files are deleted from this folder. An extra node output Deleted Files is created when you select Save. The workable is sent to this output. The workable contains a URL, which is the URL of the deleted file, but be aware that the file itself does no longer exist.
- Workable name: here you can specify the name to assign to the created workable. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- **Jacket name**: here you can specify a *jacket* name that will be searched. If found, the new workable will be linked to the jacket. If not found, a new jacket will be created, and the specified name will be assigned to it. Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.
- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.
- User: here you can specify the **created by** user to assign to the created workable. Select the pencil to open Expression Builder.

#### Start From KIOSK

With this node you can trigger the creation of a workable and/or jacket by submitting a form in KIOSK.

It has the following configuration options:

- Name: here you can specify the name that has to be used as a label by web pages to reach this input node.
- Allow Access to Scopes: here you can specify which scopes are allowed to use this input. Options:
  - If you leave this field **blank**, users with the same scope as the Whitepaper are allowed to use this input.
  - If you enter \*, all users are allowed to use this input.
  - If you enter **a comma separated list of scopes**, users that are in one of the specified scopes are allowed to use this input.
- Allow Access to Users with Attributes: here you can define which attributes users need to be allowed to use this input. Options:
  - If you leave the parameter field black, all workflows will be accessible for all users in KIOSK.
  - If you define a comma separated list of attributes, all users who have one or more of the specified attributes are allowed to use this workflow in KIOSK.
  - **Note:** Only if these users also have the correct scope as configured in Allow Access to Scopes, they will be able to access the workflows in KIOSK.
- Enable for guest users: if you select this checkbox, guest users can use this node to create workables.
- **Category**: here you can mark inputs for a certain use case. Options:
  - If you leave this field blank, no category is assigned.
  - If you enter **Approval**, you can create a custom web form to customize the fields of **Approval** tab in your **Assets**. See **APPROVAL** on page 66 for more information.
  - If you enter **Jobs**, you can build a custom web form to customize the fields when creating a new Job.
- **Kiosk parameters**: here you can configure the parameters for the custom form that is displayed in KIOSK when submitting a job to a workflow. This form fields needs to be filled in by the KIOSK user. Select the pencil to open Expression Builder.

See Form Builder on page 260 for more information.

- Use Job Form: if you select this checkbox, a Job Form is used to create some initial job data. The created data will be assigned to a single variable which can be used in the **Create Job** node.
- Job Form Name: here you can define the name of the Job Form that you specified in the Name field in the Handle Form on page 250 node.
- Job Data Variable Name: here you can define the name of the variable where the created data of the Job form will be assigned to.

- **Number of files**: in this drop-down list you can specify if users can upload files to a workflow in KIOSK. Options:
  - None: if you select this option, users cannot upload any file.
  - One: if you select this option, users *can* and *must* upload exactly one file.
  - One Or More: if you select this option, users need to upload at least one file.
- Handle files separately: if you select this checkbox, a separate workable will be created for each input file.
- or just out of convenience to not have to submit all files one by one. It might be better to process all the files in separate workables if you just submit multiple files out of convenience. Select this option to get this behavior.
- Enable File store browsing: if you select this checkbox, users are able to select files by browsing through the files and folders in the File stores.
- Enable upload: if you select this checkbox, users are able to upload files.
- Upload Location: here you can indicate where the uploaded files are stored on the server before they enter the flow. Each file will be stored in a unique folder inside the specified location and needs to be moved into the correct location by the flow if needed.



**Note:** The upload location must use the exact notation as on the file system. If not, upload from Kiosk will fail in order to prevent problems when rerunning the Jacket.

• Upload Timeout: here you can specify how long an uploaded file will be kept before it is removed in case it is not used for a real submit.

## Example

Suppose you upload one ore more files in KIOSK. However, for some reason you do not submit anything (for example because the connection was broken). In that case, the uploaded files are on the server but will never be used. With this time-out you can specify after how long such files are removed from the server.

- Max File Size for (guest) upload: here you can specify the maximum file size (in MB) of the files that guest users can upload. For regular users, no limit is defined.
- Workable name: here you can specify the name to assign to the created *workable*. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- Link To Jacket (by name): here you can specify a *jacket* name that will be searched.
  - If found, the new workable will be linked to the jacket.
  - If not found, a new jacket will be created, and the specified name will be assigned to it.

Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.

- Priority: in this drop-down list you can specify the priority to assign to the created workable.
- Scope From User: if you select this checkbox, the scope for the workable is taken from the user that is creating the workables. If you do not select it, the scope from the selected Whitepaper will be used.

## Start From Mail

With this node you can create a workable and/or jacket with an email.

It has the following configuration options:

- **Mail Server**: here you can select the mail server that will be monitored to check for new mails. Both IMAP and pop3 are supported, with or without security.
  - **pop3:**//: this will select a pop3 account without security.
  - **pop3s:**//: this will select a pop3 account with security.
  - imap://: this will select a IMAP account without security.
  - imaps://: this will select a IMAP account with security.
- Mail User: here you can select the user name to log in on the mail server.
- Mail Password: here you can select the password to log in on the mail server. Passwords are encrypted.

- **Mail Authentication Type**: here you can select the mail authentication type for the mail server. You only need to change this if the mail server does not support automatic authentication. Options:
  - Automatic: if you select this option, the mail server indicates which authentication it supports.
  - LOGIN: if you select this option, the mail server uses LOGIN authentication.
  - **PLAIN**: if you select this option, the mail server uses PLAIN authentication.
- Enabled: only if you select this checkbox the mail server will be monitored.
- Check Interval: here you can specify how often you want to monitor the mail server.
- Location To Save Attachments: here you can specify where the message body and optional attachments will be stored on the server before they enter the flow. All files from a single mail will be stored in a unique folder inside the specified location and need to be moved into the correct location by the flow if needed.
- Workable name: here you can specify the name to assign to the created *workable*. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- **Jacket name**: here you can specify a *jacket* name that will be searched. If found, the new workable will be linked to the jacket. If not found, a new jacket will be created, and the specified name will be assigned to it. Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.
- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.
- User: here you can define the created by user to assign to the created workable.

## **Start From Timer**

With this node, a *workable* is created when the configured times reaches its next execution time.

It has the following configuration options:

- **Day of month**: here you can specify the days of the month when the timer should run. You can select to run the timer on all days, or on one specific day of the month. The day of the month can also be specified **relative** to a the start or end of the month.
- Optionally limit the day to certain days of the week: here you can select checkboxes to limit certain days of the week when the timer should run.

## Examples

- If you select **Run every day** and you select the checkboxes **Mon** and **Tue**, the timer will run every Monday and Tuesday.
- If you select **Run each first day of the month** and you select the checkboxes **Mon** and **Tue**, the timer will run on each first enabled day, so each Monday or Tuesday of the month.
- If you select **Run each last day of the month** and you select the checkbox **Fri**, the timer will run on the last Friday of the month.
- How late should the timer fire the first time on active day: here you can specify the time of day the timer should run. For timers where repeat is enabled: this time is the first time the timer will run on a day the timer must run.
- **Repeat Mode**: here you can specify how the timer should repeat. Options:
  - Fire only once: there will be no repeating.
  - Repeat rest of day: the timer will stop the next day. The timer will only run that day.
  - **Repeat a fixed number of time**: the timer will fire exactly as many times as you select. The timer might fire less often if there is not enough time in the day (after the first fire) to fire the specified number of times.
- How much after the first fire should the timer fire again: here you can specify the time between two fires of the timer on the same day.
- Enabled: if you select this checkbox, the timer will be enabled.
- Workable name: here you can specify the name to assign to the created workable. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- Jacket name: here you can specify a *jacket* name that will be searched. If found, the new workable will be linked to the jacket. If not found, a new jacket will be created, and the specified name will be assigned to it. Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.

- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.
- User: here you can specify the **created by** user to assign to the created workable. Select the pencil to open Expression Builder.

## **General Workflow Constructs**

### **Call Sub Flow**

With this node you can call a sub flow in a (different) White Paper.

It has the following configuration options:

- White Paper Name: here you can define the name of the workflow where the sub flow is found. If you leave this field empty, the system searches for the Start Sub Flow node in the same whitepaper as this node.
- Sub Flow Name: here you can define the name of a sub flow as specified in the parameters in the Start Sub Flow node.
- Enable Hook: when checked the sub flow allows the use of Hook Identifiers for it to be embedded in other work flows.
- Hook Identifier: enter a unique ID that will call this sub flow to be used in other work flows. This ID is listed in the Manage hooks tab.
- File to process in Sub Flow: here you can define the file(s) to use as input files in the sub flow. Select the pencil to open Expression Builder.
- Variables For Sub Flow. Options:
  - All Variables from workable: if you select this option, all variables from the workable will be returned.
  - **Define Variables**: if you select this option, you can specify some additional parameters that will be returned. Select + to add a variable and define a **Name** and a **Value** for the variable.
- **Return Variable Name**: here you can specify the name of the variable to store the variable(s) returned by the called sub flow. If you set this name to an empty string, the returned variables are saved directly at the top level of the variables in your *workable*.

### **Related concepts**

#### Start Sub Flow on page 152

This flow indicates a starting point of a flow-path that can be used from other flows. At the end, the result of the flow will be sent back to the calling flow.

#### Return From Sub Flow on page 147

This node is the end of a flow-path where the workable is sent back to the node where the sub flow was called.

## **Check File**

With this node you can check if a file exists or does not exist, and optionally wait for the file to appear or disappear.

It has the following configuration options:

- File to check: here you can specify the file to check. Select the pencil to open Expression Builder.
- Check if exists: here you can specify if you want to check if a file exists or does not exist. Enable this option to check for existing files, disable this option to check for not-existing files.
- Wait for match: here you can specify if you want to wait for the check to succeed. Enable this option to wait for success, or disable this option to send the *workable* to a connector once the test has been done.
- Delay: here you can specify the number of seconds to wait if Wait For Match is enabled.

## **Clone Workable**

With this node you can create an exact copy of the *workable*, for example to do parallel processing.

Normally all steps of a flow are executed in sequential order when a workable passes through the nodes of a flow. However, you can execute several nodes at the same time with the **Clone Workable** node. The workable that enters the node will be copied one or several times and includes the complete history. The next nodes (regardless whether they process the original workable or the cloned workable) will see exactly the same information for all workables. All workables follow a completely independent path. At the end you have two choices:

- 1. You are not interested in the history of the cloned workables. In that case, you can end the cloned path(s) with the **End** node.
- 2. You want to continue at a certain point in your flow when all cloned workables are processed. In that case, you need to use the Join Workables node, where the original workable and all cloned workables come back together. All the workables will wait in that node until all of them are ready. The information from all workables is combined in the original workable. All cloned workables will be removed from the system and the original workable continues.

The node has the following configuration options:

• **Clones**: here you can specify the names of the output connectors to where a clone/copy of the workflow will be send. You can specify more than one output connector.

**Note:** Parallel processing is only useful when you have a system that can handle several nodes at the same time.

#### **Related concepts**

#### Join Workables on page 147

With this node you can wait for a set of workables and continue when they are all available.

#### End

With this node you can indicate the end of a flow-path. A *workable* that arrives in an end node will be marked as **finished**, and it will be removed from the system. From that point, you can only access the job history via the *jacket*.

It has the following configuration options:

- **Delay**: here you can specify how long the workable will stay in this node before it is removed from the system. From that point, you can only access the job history via the jacket.
- **Delete jacket**: if you select this checkbox, the jacket linked to the workable will be deleted when the workable is removed at the end of the flow path.
  - Attention: Be very careful with this option. The consequence of removing the jacket is that you will never find any history of the jacket and related workable(s) again in CLOUDFLOW. Everything is removed after deleting the jacket.

#### Form input

With this node you can specify that the *workable* and/or *jacket* should wait for the user to submit a form before automatically continuing.

It has the following configuration options:

- URL: here you can specify the URL that will be returned if a web-front-end asks which URL should be displayed for this workable and/or jacket. Leave this parameter empty if you don't want to change or specify an URL to display. Select the pencil to open Expression Builder.
- **Timeout**: here you can specify how long a workable should wait before automatically continuing.

## Get User Info

With this node you can receive information of/for a user (for example name, email address...)

It has the following configuration options:

- User to get info for: the user you want to get information of/for. Select the pencil to open Expression Builder.
- Variable Name: here you can specify the name of the variable where the user information will be stored. The default is user\_info.

#### Go To Input

With this node you can continue working in another workflow while keeping the complete history and information of the *workable*.

It has the following configuration options:
- White Paper Name: the name of the workflow where the workable should move to.
- Input Name: the name of an input as specified in the parameters of the Start From Web Request or Start From KIOSK nodes.

### Hold in Kiosk

With this node you can put a workable on hold in the workflow.

The workable stays on hold until a user intervention takes place. This intervention releases the workflow.

The node has the following configuration options:

• **Kiosk parameters**: here you can configure the parameters for the custom form that is displayed in KIOSK. This form fields needs to be filled in by the KIOSK user. Select the pencil to open Expression Builder.

See Form Builder on page 260 for more information.

- Use Job Form: if you select this checkbox, a Job Form is used to add new or edit existing job data.
  - If the values were already in the initial form data, they can be edited in the form.
  - If the values were not in the initial form data, the new values will be added.

The created data will be assigned to a single variable which can be used in the Create Job node.

- Job Form Name: here you can define the name of the Job Form that you specified in the Name field in the Handle Form on page 250 node.
- Job Data Variable Name: here you can define the name of the variable where the created data of the Job form will be assigned to, or where they are read from (in case they already exist).
- Editable files: here you can specify one or more files that the user can edit. You can leave this parameter empty if the user is not allowed to edit any files while the *workable* is on hold. Select the pencil to open Expression Builder.
- Viewable files: here you can specify one or more files that the user can view. You can leave this parameter empty if the user is not allowed to view any files while the workable is on hold. You can specify a folder URL. If you have CLOUDFLOW Plug-in Suite on page 419 installed, you can select this link in Illustrator, which opens the folder in the Finder/File Explorer. Select the pencil to open Expression Builder.
- Use File Store Access: if you select this checkbox, you can edit the file directly on the File store. If you leave the checkbox unselected, the file is downloaded to the local folder configured in CLOUDFLOW PLUGIN-IN SUITE SETTINGS > Work Folder and you can edit the file from this location.
  - =

**Note:** This checkbox is only useful in case of a CLOUDFLOW Plug-in Suite setup. See CLOUDFLOW Plug-in Suite on page 419 for more information.

- Enable this option if you have a setup where the editable files can be edited directly on the file store (for example in PACKZ). In this case no upload/download is needed, and CLOUDFLOW will check if files are edited when the workable continues.
- File Upload: here you can specify if a user can upload additional files in KIOSK. Options:
  - **Disabled**: if you select this option, the user cannot upload additional files.
  - Enabled, One: if you select this option, the user can and must upload exactly one additional file.
  - Enabled, One Or More: if you select this option, the user can upload one or more additional files, but at least one.
  - Enabled, None or One: if you select this option, the user can upload one or none additional files.
  - Enabled, None, One Or More: if you select this option, the user can upload any number of additional files.
- Upload Location: here you can indicate where the uploaded files are stored on the server before they enter the flow. Each file will be stored in a unique folder inside the specified location and needs to be moved into the correct location by the flow if needed.

• Upload Timeout: here you can specify how long an uploaded file will be kept in the upload folder before it is removed if it is not used for a real submit.

#### Example

Suppose you upload one ore more files in KIOSK. They are stored in the folder you specified in **Upload Location**. However, for some reason you do not submit anything (for example because the connection was broken, or you decide to do something else). In that case, the uploaded files are on the server but will never be used. With this time-out you can specify after how long such files are removed from the server.

- **Timeout**: here you can specify how long a workable should wait before automatically continuing.
- **Routing**: here you can specify the possible decisions for the user. Each entry corresponds with a specific output connector. Each entry will be visible for the user in KIOSK. When the user makes a decision, the workable is released.
- **Default Connector**: here you can select the connector that will be selected if the workable continues because the time out was reached.

### House Keeping

With this node you can change various settings of a *workable*.

It has the following configuration options:

- Activity: here you can select the setting to change. Options:
  - Set Name: if you select this option, the name of the *workable* and/or *jacket* is changed.
  - Set Role: if you select this option, the role in the workable is updated.
  - Set Workable State: if you select this option, the state in the workable is updated.
  - Set Data Access Scope: if you select this option, the scope that will be used when accessing data (such as assets) in the workable is set.
  - Set Data Priority: if you select this option, the priority in the workable is set.
  - Set Work Server: if you select this option, the work servers were this workable can be ran on are limited.
  - Set Logging Level: if you select this option, the amount of log information that is kept in the workable and/or jacket is reduced.

### When Activity is set to Set Name

- Name: here you can specify the new name to assign to the workable and/or Jacket.
- Set Name of Workable: if you select this checkbox, the name of the workable is changed.
- Set Name of Jacket: if you select this checkbox, the name of the jacket is changed.

### When Activity is set to Set Role

- Type: here you can specify the role to set/change. Options:
  - Handler: if you select this option, the role is updated to the person that is responsible to handle the workable.
  - Creator: if you select this option, the role is updated to the person that created the workable.
  - **Operator**: if you select this option, the role is updated to the person that last made a workflow decision on the workable. For example, this is the person that made a decision on a workable in **Hold In Kiosk**.
- User Name: here you can specify the person to assign to the selected role. Select + to assign various users. Select the pencil to open Expression Builder.
- Attribute: here you can specify the attribute that select the persons to assign to the selected role.

### When Activity is set to Set Workable State

- Error state: here you can specify the new state to assign to the workable.
- **Message**: here you can specify the text that will be visible for the user. You can for example specify why the job goes in error state.

### When Activity is set to Set Data Access Scope

• Scope Name: here you can specify the scope to assign.

### When Activity is set to Set Priority

• **Priority**: here you can specify the priority to assign.

### When Activity is set to Set Work Server

• Work Server to Use: here you can specify the work server to run on. If you leave this field empty, the workable is allowed to run on all work servers again after limiting to a specific work server.

### When Activity is set to Set Logging Level

- Logging Level: here you can specify the amount of log information that is kept in the workable and/or jacket. Read the possibilities from top to bottom: each line below another removes the documented data when compared to the line above. Options:
  - Full: if you select this option, the standard detailed log information is kept.
  - Limited Parameters in Jacket History: if you select this option, *large* parameters used during processing of a node from the jacket are removed.



Note: Large parameters are nested JSON objects of more than 2 levels.

- No Parameters in Jacket History: if you select this option, the parameters used during processing of a node from the jacket are removed.
- Limited Parameters in Workable History: if you select this option, *large* parameters used during processing of a node from the workable are removed.

#### Join Workables

With this node you can wait for a set of workables and continue when they are all available.

A set of workables belong together and are ready under the following conditions:

- Only workables that are linked to the same *jacket* can be combined.
- Each input connector must contain a workable that belongs to the same jacket.

This also means that if you only have one input connector, each incoming workable will be sent immediately to the success output connector. If you want to collect the results after applying **Clone Workables**, you need to create a **Clone Workables** node where the number of input connectors is equal to the number of output connectors of the **Clone Workables** node.

The node has the following configuration options:

- **Data**: here you can specify what you want to merge/join. If you don't change this field, it will merge everything that comes. Only change this if you are really need it.
- Routing: here you can specify all the names of the input connectors. This way the Clone Workables can enter.

#### **Return From Sub Flow**

This node is the end of a flow-path where the workable is sent back to the node where the sub flow was called.

- File to return: here you can define the file(s) to return as results files in the calling flow. Select the pencil to open Expression Builder.
- Variables to return: here you can define the variables to return. Options:
  - All Variables from workable: if you select this option, all variables from the workable will be returned.
  - **Define Variables**: if you select this option, you can specify some additional parameters that will be returned. Select + to add a variable and define a **Name** and a **Value** for the variable.

• **Return Failure**: if you select this checkbox, you can control the output path of the node where the sub flow was called from. Normally the **success** path is used, but if you select this checkbox the **failure** path is used. For example, in case of an error in the sub flow, you can end the error handler path by adding the **Return Form Sub Flow** node and selecting this checkbox.

#### **Related concepts**

Call Sub Flow on page 143 With this node you can call a sub flow in a (different) White Paper.

### Start Sub Flow on page 152

This flow indicates a starting point of a flow-path that can be used from other flows. At the end, the result of the flow will be sent back to the calling flow.

### Repeat

With this node you can repeat a section of a flow one ore more times.

Repeating can be based on:

- A number that determines the count.
- The files where the repeat will then select each individual file once.
- A JSON array where the repeat will then select each item in the array once.

The node has the following configuration options:

- Mode: here you can specify what needs to be iterated/repeated. Options:
  - **Counter**: if you select this option, the **NR Repeats** parameter will determine how many times the repeat flow will be called.
  - Enumerate Files: if you select this option, the files referenced by the **Data** parameter will be enumerated. Using **Data from Previous Node** in the repeat flow will select each time one of the referenced files. At the end of the Repeat, all resulting files from the repeat flow will be collected and set as output files of the Repeat Node itself.
  - Enumerate JSON Array: if you select this option, the data referenced by the Data parameter will be enumerated. That referenced data must be an Array. Using Data from Previous Node in the repeat flow will select each time one of the array entries (if the 'Select Sub JSON' option is enabled). At the end of the Repeat, all resulting data from the repeat flow will be collected and set as output data of the Repeat Node itself (if the Update JSON option is enabled).
- **Create Multiple Workables**: if you select this checkbox, one workable is created for each iteration, where all those workables will process independently until the repeat is finished. If you leave this checkbox unselected, the repetition is performed on the single *workable*.
- Variable name: here you can specify the name of the variable to store the repeat information in. The repeat information contains:
- • The number of times the repeat section will be used.
  - How many times the section was already repeated (the index). This value will be zero the first time the repeat section is used.

You can set it to an empty string if you don't need this information.

When Mode is set to Counter:

• Nr Repeats: here you can specify the number of times you want to repeat.

#### When Mode is set to Enumerate Files or Enumerate JSON Array:

• **Data**: here you can specify the data to use while repeating. This can either be a list of files, or it can be JSON data. Select the pencil to open Expression Builder.

When Mode is set to Enumerate JSON Array:

- Select Sub JSON: here you can specify if you want the **Repeat Node** to set the selected part of the JSON array that is being iterated as its output data during Repeat. Enable this option to set the data, or disable this option if you do not need this data.
- Update JSON: here you can specify if you want the **Repeat Node** to update the JSON array with the output data or the Repeats. Enable this option to update the data, or disable this option if you want to keep the iterated JSON array unchanged. This option can not be used if the iterated JSON array is coming from a variable or if the iterated JSON array is build on the fly.

### Route

With this node you can route a job to one output depending on information in the workable, file or data.

- Value: here you can specify the information to compare (also called **Parameter**). Select the pencil to open Expression Builder.
- Value Type: here you can specify how to interpret the Parameter value before comparing. Options:
  - Use data without interpretation: if you select this option, the value will be used as found. This means:
    - If you enter the value to compare in the edit field, it will be used as text.
    - If you receive the value from a variable, the data type will be the type of the contents of the variable.
  - Text: if you select this option, the value will be converted to text. This means:
    - the data will be used without modifications if it is text already (for example when you enter it in the edit field), or
    - it is converted to text if it is a number or true or false, or
    - it will be converted to a JSON String if it is an JSON array or JSON object.
  - **Number**: if you select this option, the **value** will be converted to a number. Note that the decimal separator must be a point.
  - **True or False**: if you select this option, the **value** will be converted to **true** or **false**. This means that the string **true** (regardless of upper, lower or mixed case) will be converted to **true**, numbers with a non-zero value will be converted to **true**, and all other data will be converted to **false**.
  - JSON Data: if you select this option, the value will be interpreted as a JSON String, and converted to data will be used for comparison. This can be used to compare several values at once.
- Test: here you can specify how to compare the Parameter with the selected value(s). Options:
  - Equal to: if you select this option, the parameter and the specified value must match exactly.
  - Less than: if you select this option, the parameter must be less than the specified value.
  - Less than or equal to: if you select this option, the parameter must be equal to or less than the specified value.
  - Greater than: if you select this option, the parameter must be greater than the specified value.
  - Greater than or equal to: if you select this option, the parameter must be equal to or greater than the specified value.
  - Contains: if you select this option, the parameter must contain the specific value.
  - Begins with: if you select this option, the parameter must begin with the specified value.
  - Ends with: if you select this option, the parameter must end with the specified value.
- Routing
  - **Compare to**: here you can specify the value to use when comparing with the Parameter.
  - Label: here you can specify the output connector where the flow must go when the specified value matches up with the Parameter.

- Routing Value type: here you can specify how to interpret the compare value(s) before comparing. Options:
  - Use data without interpretation: if you select this option, the value will be used as found. This means:
    - If you enter the value to compare in the edit field, it will be used as text.
  - If you receive the value from a variable, the data type will be the type of the contents of the variable.
  - Text: if you select this option, the value will be converted to text. This means:
    - the data will be used without modifications if it is text already (for example when you enter it in the edit field), or
    - it is converted to text if it is a number or **true** or **false**, or
    - it will be converted to a JSON String if it is an JSON array or JSON object.
  - **Number**: if you select this option, the **value** will be converted to a number. Note that the decimal separator must be a point.
  - **True or False**: if you select this option, the **value** will be converted to **true** or **false**. This means that the string **true** (regardless of upper, lower or mixed case) will be converted to **true**, numbers with a non-zero value will be converted to **true**, and all other data will be converted to **false**.
  - JSON Data: if you select this option, the value will be interpreted as a JSON String, and converted to data will be used for comparison. This can be used to compare several values at once.
- **Compare Number Accuracy**: here you can specify the possible rounding when comparing numbers. All numbers that do not differ more than the value specified here will match when compared for equality.
- Case Free String Compare: if you select this checkbox, strings must be compared case free.

### Script

With this node you can run a JavaScript.

It has the following configuration options:

- Input Files: here you can specify the file that will be presented to the script for processing.
  - Note: By default, the Input files parameter is set to files from previous node. In general this is OK, but it can lead to unexpected problems when you use a Script node immediately after a Start From KIOSK node that does not expect any input files. In that case, there are no previous files and the Script node will fail. The solution is to clear the Input Files parameter.

Select the pencil to open Expression Builder.

- Variables For Script: here you can define the variables for the script. Options:
  - All Variables from Workable: if you select this option, the variables from the workable will be used.
  - **Define Variables**: if you select this option, you can specify some additional parameters that will be presented to the script. Select + to add a variable and define a **Name** and a **Value** for the variable.
- **Output connectors**: here you can specify additional output connectors. From the script you can select any output you want/need, including these additional outputs.
- Log function feedback: with this checkbox you can specify if the feedback from called API functions will be added to the *workable* Log. If you check this checkbox, all messages logged by API functions will be returned as a result from the calls, and they will be added to the log of the workable. If you leave this checkbox unselected, any message logged by API functions will only be returned as a result from the calls.
- Fail on function errors: with this checkbox you can specify if the script node will error in case one of the called functions reports an error. When you select this checkbox, the script will fail immediately when a function returns an error, and the workable will be sent to the failure output. This way you can start without error handling in your script, and still do the right thing when a function fails. When you leave this checkbox unselected, it is the script's responsibility to react correctly when a function returns an error.
  - **Note:** Log Function Feedback and Fail On Function Errors are enabled by default, because it helps during initial Script setup. Initially you probably do not add error handling, and you want to see all possible feedback to find out where you made mistakes in your script. Later, when your script evolves, you might decide to disable one or both options, depending on the functionality and/or error handling you add to the script.

• **Max Concurrent Workers**: here you can specify how many concurrent calls to the script can be executed. The script itself normally has no limits, but you might want to limit the maximum concurrency to not completely lock your system on this node if you perform slow calls or long running calls to other components from your script.

If you set it to 0, the concurrency is not limited. If you set it to 1, this node will only be executed once at the same time. If you set it to 2, this node might be executed at most 2 times at the same time, etc.

• Script File: here you can run a script from a file. The script specified in the field below will be ignored when this

parameter is specified. Select 🔡 to expand the script editor to the full dialog.

See Functions in the Script node on page 464 for more information on which functions you can use in the Script node.

#### Set DBIO Reply

With this node you can return the answer that was collected in the workable that was created in the **Start From DBIO** node in the same workflow.

It has the following configuration options:

- **Result Mode**: here you can select the result mode. Options:
  - Success: if you select this option, the collected data is returned. You can define the data in the field Contents.
  - The requested document was not found: if you select this option, an error is returned. You can define the error message in the field Error Description.
- Contents: here you can define the collected data to return, in case the Result Mode is set to Success.
- Error Description: here you can define the error description, in case the Result Mode is set to The requested document was not found.

See also Start From DBIO on page 137.

#### Set HTTP Reply

With this node you can set the reply that should be sent as an answer to an HTTP request.

It has the following configuration options:

- Status: here you can define the HTTP status code to return to the caller. For example, if you set the status code to 303, a page in PAGEBULDER is redirected if the workable does not exist.
- **Content type:** here you can define the mime type to return to the caller for the page contents.
- Contents: here you can define the page contents to return to the caller.
- **Delay**: here you can specify the amount of time that the *workable* will wait before continuing to the next node. This is important if the next node is an **End** node: in that case the complete system needs the time to read the reply, and that is only possible if the workable remains active long enough.

#### Set Kiosk Info

With this node you can add/set information to be displayed in KIOSK.

It has the following configuration options:

- Activity: here you can select what to set. Options:
  - Set Files: if you select this option, the specified files are added to the list of result files in KIOSK.
  - Set Data: if you select this option, a value is added/set to be displayed in KIOSK.
  - Set File For Preview: if you select this option, the file to be used in KIOSK to display a preview of the *workable* is set.
  - Set File For Jacket Preview: if you select this option, the file to be used in KIOSK to display a preview of the *jacket* is set.
  - Reset Files: if you select this option, the files in the Kiosk info are reset.
  - Reset Data: if you select this option, the data in the Kiosk info are reset.

When Activity is set to Set Files:

• Files: here you can specify the file(s) to add. Select the pencil to open Expression Builder.

When Activity is set to Set Data:

- Display Name: here you can specify the name of the data.
- Data Type: here you can specify the data type (Text, Hyperlink, Number, Number with decimals).
- Stored As: here you can specify how the data will be stored.
- Value: here you can specify the value.

When Activity is set to Set File For Preview:

• File for Preview: here you can specify the file that will be used as the preview for the workable.

When Activity is set to Set File For Jacket Preview:

• File for Preview: here you can specify the file that will be used as the preview for the jacket.

### Set Server Page

With this node you can change the Web Page that a user will view.

It has the following configuration options:

• URL: here you can specify the URL that will be returned if a web-front-end asks which URL should be displayed for this *workable* and/or *jacket*. Select the pencil to open Expression Builder.

### Set Variable

With this node you can introduce one or more variables in your workflow and set the value (which can be based on an existing variable).

It has the following configuration options:

- Name: here you can specify the name of the variable to set. Select the pencil to open Expression Builder.
- Value: here you can specify the value to set.

Select + to add a variable.

**Note:** When you set a variable, the variable will be available for all *workables* belonging to the same *jacket*.

#### **Start Sub Flow**

This flow indicates a starting point of a flow-path that can be used from other flows. At the end, the result of the flow will be sent back to the calling flow.

It has the following configuration options:

- Name: here you can specify the name that has to be used to call this starting point.
- Input parameters: in this table you can define the parameters that need to be specified by callers from this flow.
- Check Input parameters: if you select this checkbox, the input parameters are checked, and only the ones defined in the **Parameter List** will be copied and used. Also, the Sub Flow will not be started if a required parameter is not specified. This removes the need to handle missing parameters in your sub Flow. However, in some cases (for example when you use the Sub Flow as a mechanism to simplify a single large workflow), your Sub Flow will only be called from one place, and you want access to any/all variables in the calling workflow. If you leave this checkbox unselected, the input parameters are not checked.

### **Related concepts**

#### Call Sub Flow on page 143

With this node you can call a sub flow in a (different) White Paper.

#### Return From Sub Flow on page 147

This node is the end of a flow-path where the workable is sent back to the node where the sub flow was called.

#### **Unhandled Problem**

All *workables* that fail in a node that are not explicitly handled in a flow will be routed to this node. Each workflow has by default an unhandled problem node.

- **Delay**: here you can specify how long the system needs to wait to remove the workable. From that point onwards, the job history can only be accessed from the *jacket*.
- Enable Handler: if you select this checkbox, workables that are routed to this node should be processed by other nodes (for example to do some cleanup).

### Wait

With this node you can pause the *workable* for a specified amount of time.

It has the following configuration options:

• Delay: here you can specify the amount of time to wait.

# **File Handling**

### **Compare Files**

With this node you can compare one or more files with a set of reference files.

It has the following configuration options:

- File to compare: here you can specify the file(s) to compare. Select the pencil to open Expression Builder.
- File/Folder to compare with: here you can specify the file or folder to compare with.
  - If you specify a **folder**, each file will be compared with the file in this folder that has the same name. If a file with the same name cannot be found in the folder, the operation will error.
  - If you specify a file, the source file will be compared with this specific file.
- Strict binary compare: if you select this checkbox, the files will be compared on a binary level. If you leave this checkbox unselected, the files will be compared based on their content.
  - PDF files and images will be visually compared.
  - Text files will be textually compared.
  - JSON files will be compared as JSON objects.
- **Difference file (visual compare)**: here you can define a location where a difference TIFF file will be written in case the difference file is not empty and the visual compare fails. It will help identifying all the differences.

### **Copy File**

With the this node you can copy one or more files to another location.

- File to copy: here you can specify which file(s) should be copied. Select the pencil to open Expression Builder.
- To file or folder: here you can specify the file or folder to copy to. If you specify a folder, then the file will be copied with the original name into that folder. If you specify a file, then the file will be copied exactly to the specified name. Select the pencil to open Expression Builder.
- **Create folders**: if you select this checkbox, all folders that are needed to copy the file will be created. If you leave this checkbox unselected, copying the file will fail if you try to copy to a folder that does not exist (for example when you generate **to folder** on the fly).
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- Unique name mode: here you can control the naming of the copied file. Options:
  - None: if you select this option, the same name as the input file will be used.
  - Sequential: if you select this option, the specified name will be used if the default copied file does not exist. However, if it does exist, the system will try to add 1 to the file name, and check if that file exists. If it does not exist, the updated file name is used. If it does exist, 2 is tried, etc...
  - **Random**: if you select this option, the specified name will be used if the **default** copied file does not exist. However, if it does exist, the system will try to add a random suffix to the file name, and check if that file exists. If it does not exist, the updated file name is used. If it does exist, a new random suffix is tried, etc... Use

this mode if you just want a unique name (for example for a temporary file) and you don't care about the actual name.

- Sequential+Always: if you select this option, the same as Sequential but it always appends a number, even if the default copied file does not exist.
- **Random+Always**: if you select this option, the naming will follow the same rules as when the **Random** option is selected, but it always appends a suffix, even if the **default** copied file does not exist.

### Copy Folder

With the this node you can copy one or more folders to another location.

It has the following configuration options:

- Folder to copy: here you can specify which folder or folder content should be copied. Select the pencil to open Expression Builder.
- To file or folder: here you can specify the folder to copy the folder of folder content to. Select the pencil to open Expression Builder.
- **Only copy contents**: if you select this checkbox, only the content of the folder is copied. If you leave this checkbox unselected, both the folder itself and the content are copied.
- Create folders: if you select this checkbox, all folders needed to copy the folder will be created. If you leave this checkbox unselected, copying to a folder that does not exist will fail (for example when you generate to folder on the fly).
- **Overwrite mode**: here you can select the required behavior when files and folders already exist in the **To Folder**. Options:
  - **DoNotReplaceRootFolder**: if you select this option, the folder copy will fail if the specified **To Folder** already exists (and **Unique Name Mode** is set to **None**).
  - **ReplaceRootFolder**: if you select this option, the specified **To Folder** is deleted first (in case it already exists), then the copy is executed.
  - **MergeTreeAndDoNotReplaceFiles**: if you select this option, the **To Folder** will contain the merged result of the **Folder To Copy** and an already existing folder (if applicable). If the **To Folder** does not yet exist, this works exactly like **ReplaceRootFolder**. If the **To Folder** does exist, the files in the **Folder To Copy** will only be copied if there is no corresponding existing file in the **To Folder**.
  - **MergeTreeAndReplaceFiles**: if you select this option, the **To Folder** will contain the merged result of the **Folder To Copy** and an already existing folder (if applicable). If the **To Folder** does not yet exist, this works exactly like **ReplaceRootFolder**. If the **To Folder** does exist and if a corresponding file in this folder is found, it will be deleted first and the incoming file will be copied. This is the case for each file in the incoming folder.
- Unique name mode: here you can control the naming of the copied folder. Options:
  - None: if you select this option, the same name as the input folder will be used.
  - Sequential: the specified name will be used if the default copied folder does not exist. However, if it does exist, the system will try to add 1 to the folder name, and check if that folder exists. If it does not exist, the updated folder name is used. If it does exist, 2 is tried, etc...
  - **Random**: the specified name will be used if the **default** copied folder does not exist. However, if it does exist, the system will try to add a random suffix to the folder name, and check if that folder exists. If it does not exist, the updated folder name is used. If it does exist, a new random suffix is tried, etc... Use this mode if you just want a unique name (for example for a temporary folder) and you don't care about the actual name.
  - Sequential+Always: the same as Sequential but it always appends a number, even if the default copied folder does not exist.
  - **Random+Always**: if you select this option, the naming will follow the same rules as when the **Random** option is selected, but it always appends a suffix, even if the **default** copied folder does not exist.

### **Create Folder**

With this node you can create a new folder.

It has the following configuration options:

• Create new folder inside this folder: here you can specify the folder to create the new folder in. Select the pencil to open Expression Builder.

- Folder to create: here you can specify the path of the folder to create inside the specified folder. This path should always end with a slash (/). You can specify a path with multiple sub-folders. If the requested folder already exists, no action will take place. Select the pencil to open Expression Builder.
- Create folders (in "inside" folder): if you select this checkbox, the folders that are specified in the parameter field Create new folder inside this folder are created before the folder that is specified in the parameter field Folder to create is created. This options prevents that the node will fail if you try to create a folder inside a folder that does not exist.
- Unique name mode: here you can control the naming of the copied folder. Options:
  - None: if you select this option, the same name as the input folder will be used.
  - Sequential: the specified name will be used if the **default** copied folder does not exist. However, if it does exist, the system will try to add 1 to the folder name, and check if that folder exists. If it does not exist, the updated folder name is used. If it does exist, 2 is tried, etc...
  - **Random**: the specified name will be used if the **default** copied folder does not exist. However, if it does exist, the system will try to add a random suffix to the folder name, and check if that folder exists. If it does not exist, the updated folder name is used. If it does exist, a new random suffix is tried, etc... Use this mode if you just want a unique name (for example for a temporary folder) and you don't care about the actual name.
  - Sequential+Always: the same as Sequential but it always appends a number, even if the default copied folder does not exist.
  - **Random+Always**: f you select this option, the naming will follow the same rules as when the **Random** option is selected, but it always appends a suffix, even if the **default** copied folder does not exist.

### **Delete File**

With this node you can delete one or more files.

It has the following configuration options:

- File to delete: here you can specify which file(s) to delete. Select the pencil to open Expression Builder.
- **Delete Enclosing Folder**: if this checkbox is selected, the folder that contained the deleted file will be removed as well. This is helpful when the deleted file(s) was/were initially created in a temporary folder. The folder will only be deleted in case it is empty after deleting the file(s). If you requested to delete the folder when it is not empty, a warning pops up. The node will not fail.
- **Ignore Missing File**: if you select this checkbox, the system will not generate an error in case a non-existing file is deleted. If you leave this checkbox unselected, an error will be generated in case a non-existing file is deleted.

### **Delete Folder**

With this node you can delete a folder and/or its content.

It has the following configuration options:

- Folder to delete: here you can specify which folder to delete. Select the pencil to open Expression Builder.
- **Only delete folder contents**: if you select this checkbox, only the folder's content is deleted and the folder itself is not deleted. If you leave this checkbox unselected, both the folder and its content will be deleted.
- **Delete Enclosing Folder**: if you select this checkbox, the folder that contained the deleted file will be removed as well. This is helpful when the deleted file(s) was/were initially created in a temporary folder. The folder will only be deleted in case it is empty after deleting the file(s). If you requested to delete the folder when it is not empty, a warning pops up. The node will not fail.
- Ignore Missing Folder: if you select this checkbox, there is no error when the folder to delete does not exist.

#### Get File Info

With this node you can receive information from a file (for example file size, mime type...) and save the information in a variable.

- File to get info from: here you can specify the file to receive the information from. Select the pencil to open Expression Builder.
- Variable Name: here you can specify the name of the variable to store the file info in. The default name is file\_info)

- **Support Multiple Files**: if you select this checkbox, it is possible to support data from multiple files. In this case, an array will be assigned to the specified variable, and each entry will contain the info from one file. If you leave this checkbox unselected, the data from only one file will be assigned directly to the specified variable, because the info from the second file will overwrite the info from the first file.
- **Only From File**: with this checkbox you can specify whether the file info can come from the asset info, or only directly from the file. This is can be important if you have custom file info extractors. If you select the checkbox, the file info can only come from the standard file info extractors that will check the specified file. If you leave the checkbox unselected, the file info can come from the asset (which might be updated/changed when compared to the info obtained from the file).

### Join Pages

With this node you can join multiple PDF pages into a single multipage PDF file.

It has the following configuration options:

- **PDF files to join**: here you can specify the files you want to join. The files are joined in the order they are placed in the files list. Select the pencil to open Expression Builder.
- Folder to save the merged file in: here you can specify the folder where the merged PDF file will be saved in. Select the pencil to open Expression Builder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### **Related concepts**

Split Pages on page 159 With this node you can split a multipage PDF file in groups of pages.

### Manage Pages

With this node you can manage the pages of a multipage PDF document.

It has the following configuration options:

- Input file: here you can define the file(s) to process. Select the pencil to open Expression Builder.
- File/folder to write result to: here you can define the file or folder to write the results to.
  - If you specify a folder, each file will be saved in this folder with the original name.
  - If you leave this field blank, the incoming file(s) will be overwritten.

Select the pencil to open Expression Builder.

• **Pages to delete**: here you can specify which pages to delete by entering a comma separated list or a range. You can use a comma separated list or a range. For example, **1-5,16,27** will remove page 1 to 5, 16 and 27 from the input file(s).

### Move File

With this node you can move one or more files to another location.

- File to move: here you can specify which file(s) should be moved. Select the pencil to open Expression Builder.
- To file or folder: here you can specify to which location the file(s) should be moved. Select the pencil to open Expression Builder.
- Create Folders: if you select this checkbox, all necessary folder(s) are created if they don't exist. If you leave this checkbox unselected, the node will fail if you move the file(s) to a non-existing folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

- Unique name mode: here you can control the naming of the copied file. Options:
  - None: if you select this option, the same name as the input file will be used.
  - Sequential: if you select this option, the specified name will be used if the default copied file does not exist. However, if it does exist, the system will try to add 1 to the file name, and check if that file exists. If it does not exist, the updated file name is used. If it does exist, 2 is tried, etc...
  - **Random**: if you select this option, the specified name will be used if the **default** copied file does not exist. However, if it does exist, the system will try to add a random suffix to the file name, and check if that file exists. If it does not exist, the updated file name is used. If it does exist, a new random suffix is tried, etc... Use this mode if you just want a unique name (for example for a temporary file) and you don't care about the actual name.
  - Sequential+Always: if you select this option, the same as Sequential but it always appends a number, even if the default copied file does not exist
  - **Random+Always**: if you select this option, the naming will follow the same rules as when the **Random** option is selected, but it always appends a suffix, even if the **default** copied file does not exist.
- **Delete Enclosing Folder**: if you select this checkbox, the folder that contained the moved file will be removed as well. This is helpful when the moved file(s) was/were initially created in a temporary folder. The folder will only be deleted in case it is empty after moving the file(s). If you requested to delete the folder when it is not empty, a warning pops up. The node will not fail.

#### **Move Folder**

With this node you can move a folder or its content to another location.

- Folder to move: here you can specify which folder should be moved. Select the pencil to open Expression Builder.
- To file or folder: here you can specify the folder to move the folder of folder content to. Select the pencil to open Expression Builder.
- **Only move contents**: if you select this checkbox, only the content of the folder is moved. If you leave this checkbox unselected, both the folder itself and the content are moved.
- Create folders: if you select this checkbox, all folders needed to move the folder will be created. If you leave this checkbox unselected, moving to a folder that does not exist will fail (for example when you generate to folder on the fly).
- **Overwrite mode**: here you can select the required behavior when files and folders already exist in the **To Folder**. Options:
  - **DoNotReplaceRootFolder**: if you select this option, the folder move will fail if the specified **To Folder** already exists (and **Unique Name Mode** is set to **None**).
  - **ReplaceRootFolder**: if you select this option, the specified **To Folder** is deleted first (in case it already exists), then the move is executed.
  - **MergeTreeAndDoNotReplaceFiles**: if you select this option, the **To Folder** will contain the merged result of the **Folder To Move** and an already existing folder (if applicable). If the **To Folder** does not yet exist, this works exactly like **ReplaceRootFolder**. If the **To Folder** does exist, the files in the **Folder To Move** will only be moved if there is no corresponding existing file in the **To Folder**.
  - **MergeTreeAndReplaceFiles**: if you select this option, the **To Folder** will contain the merged result of the **Folder To Move** and an already existing folder (if applicable). If the **To Folder** does not yet exist, this works exactly like **ReplaceRootFolder**. If the **To Folder** does exist and if a corresponding file in this folder is found, it will be deleted first and the incoming file will be moved. This is the case for each file in the incoming folder.
- Unique name mode: here you can control the naming of the moved folder. Options:
  - None: if you select this option, the same name as the input folder will be used.
  - Sequential: the specified name will be used if the default moved folder does not exist. However, if it does exist, the system will try to add 1 to the folder name, and check if that folder exists. If it does not exist, the updated folder name is used. If it does exist, 2 is tried, etc...
  - **Random**: the specified name will be used if the **default** moved folder does not exist. However, if it does exist, the system will try to add a random suffix to the folder name, and check if that folder exists. If it does not exist,

the updated folder name is used. If it does exist, a new random suffix is tried, etc... Use this mode if you just want a unique name (for example for a temporary folder) and you don't care about the actual name.

- Sequential+Always: the same as Sequential but it always appends a number, even if the default moved folder does not exist.
- **Random+Always**: if you select this option, the naming will follow the same rules as when the **Random** option is selected, but it always appends a suffix, even if the **default** moved folder does not exist.

### **Pitstop Preflight**

With this node you can preflight a PDF using Enfocus Pitstop Server.

It has the following configuration options:

- File to preflight : here you can specify the PDF file to preflight. Select the pencil to open Expression Builder.
- **Preflighted PDF file**: here you can specify where the preflighted (and possibly updated PDF) should be saved. If this parameter is left blank, a file with a default name (this is the original name prefixed with **Preflighted**) in the same folder as the input PDF file will be used. The used file path follows the standard input-output file name combination rules valid inside CLOUDFLOW.
- XML report file : here you can specify where the XML preflight report should be saved. If this parameter is left blank, a file with a default name (original name with extension xml) in the same folder as the input PDF file will be used. The used file path follows the standard input-output file name combination rules valid inside CLOUDFLOW.
- **PDF report file**: here you can specify where the PDF preflight report should be saved. If you this field is left blank, no PDF report will be generated. By default, the report file will be written in the same folder as the input PDF. The used file path follows the standard input-output file name combination rules valid inside CLOUDFLOW.
- **PDF report template**: here you can specify the template to use when generating a PDF preflight report. This file has an extension **.prefs**.
- **Mutator file** : here you can specify the Pitstop Server mutator file that should be used. This file has an extension .ppp.
- Generate notes: if you select this checkbox, the preflight remarks are added to the asset. As a consequence they will appear in PROOFSCOPE as notes.
- **Put preflight result in variable**: here you can specify the variable in which the interpreted preflight results are saved. This JSON information can be used for further interpretation and processing of the preflight. information.
- **Pitstop Executable**: here you can specify the location of the Pitstop executable.

#### **Prepare Asset**

With this node you can prepare the asset so that it has a thumbnail and complete metadata.

It has the following configuration options:

• File to prepare: here you can specify the file to prepare. Select the pencil to open Expression Builder.

### **Previews etc**

- Generate Thumbnail: if you select this checkbox, a thumbnail of the file is added.
- Generate Always: if you select this checkbox, the thumbnail is generated in all cases (for example if you changed the JPEG quality). If you leave this checkbox unselected, the thumbnail will not be re-generated if it is already available.
- JPEG quality: here you can specify the JPEG quality of the saved thumbnail.
- Generate (Separated) Preview: if you select this checkbox, a separated preview for the file is added.
- Generate Always: if you select this checkbox, the separated preview is generated in all cases (for example if you changed the anti alias setting). If you leave this checkbox unselected, the separated preview will not be regenerated if it is already available.
- Anti Alias: if you select this checkbox, the separated preview is generated with anti-alias.
- Enable Extra 'no update' output: if you select this checkbox, an output Not Updated is added to the node. In case the node did not update the asset, this output will be followed.

### Metadata

• Calculate Area Coverage: if you select this checkbox, the metadata for the file is completed with area coverage information. This area coverage is only by (regular) metadata processing when it is directly available in the file itself.

### Select Files

With this node you can select files in a folder or subfolder.

It has the following configuration options:

- Folder to list: here you can specify the folder to list.
- **Recursive**: if you select this checkbox, all files found in the subfolder of the specified listed folder will also be returned. If you leave this checkbox unselected, only the specified folder will be listed.
- File Name Pattern: here you can specify a regular expression to filter the file names. If this parameter is set to .\*, all files will be returned.

=

Note: You can use https://regex101.com as an online regex tester.

• Expected number of files: here you can specify the number of files you expect to match. The node will fail if the found number of files does not match.

### Set File Info

With this node you can overwrite file information (for example mime type)

It has the following configuration options:

- The file to set info on: here you can specify the file to set info on. Select the pencil to open Expression Builder.
- File info: here you can specify the file info to store. Select the pencil to open Expression Builder.

### **Split Pages**

With this node you can split a multipage PDF file in groups of pages.

It has the following configuration options:

- **PDF file to split**: here you can specify the multipage file to split. Select the pencil to open Expression Builder.
- Folder to save the split page files in: here you can specify the folder to save the split files into. Select the pencil to open Expression Builder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- Amount of pages in each output file: here you can specify the amount of pages in each split file. If you specify a value greater than 1, the node will output multipage documents, with each file containing the amount of pages specified in this field. If the total page count is not a multiple of this parameter, the last document may contain less pages.

### Related concepts

### Join Pages on page 156

With this node you can join multiple PDF pages into a single multipage PDF file.

### Unzip files

With this node you can extract files from an archived file.

- Zip file to expand: here you can select the archive to expand. Select the pencil to open Expression Builder.
- Folder to save the extracted files in: here you can specify the folder to save the extracted files in. Select the pencil to open Expression Builder.
- Use path from Zip file: if you select this checkbox, the files are saved to a path starting from the specified folder, using the path that might be stored in the zip archive. If you leave this checkbox unselected, the files are saved directly in the specified folder.

- **Overwrite existing files mode**: here you can select what should be done if the file(s) to be extracted already exist(s). Options:
  - **Do Not Overwrite**: if you select this option, unzipping will fail if the file(s) to be extracted already exist(s). No files will be extracted, unless you ignore the paths in the zip archive, and several files with the same name exist in the archive.
  - **Overwrite**: if you select this option, all existing files will be overwritten.
    - **Note:** You might miss some files if you select to ignore the paths in the zip archive, and several files with the same name exist in the archive.
  - Use Alternative Name when file already exists: if you select this option, files that already exist will be renamed (this is, a number will be appended to the name).
- Fix invalid file and folder names: if you select this checkbox, file and folder names are fixed so that they are valid on the local file system. For example, names that are too long are truncated, invalid characters are replaced by an underscore...
- Save Symbolic links from Zip file: if you select this checkbox, the (symbolic) links in the zip archive are saved. If you leave this checkbox unselected, the links are ignored.
- List all extracted files for following nodes: if you select this option, the extracted files are listed in the output. If you leave this checkbox unselected, the generated files are not added as references in the workable.

### **Related concepts**

#### Zip Files on page 160

With this node you can add create a Zip archive and add files to it.

### Update XMP

With this node you can update the XMP data in a file.

It has the following configuration options:

- File: here you can select the file you want to update. Select the pencil to open Expression Builder.
- **RDF/XML**: here you can select the RDF/XML file that contains the XMP update. Select the pencil to open Expression Builder.

#### **Zip Files**

With this node you can add create a Zip archive and add files to it.

It has the following configuration options:

- Files to add to the Zip file: here you can specify which file(s) must be added to the archive. Select the pencil to open Expression Builder.
- Folder to create the archive in: here you can specify in which folder the archive must be created. Select the pencil to open Expression Builder.
- Archive Name: here you can specify the name of the archive.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- Add Macintosh Data: if you select this checkbox, Macintosh specific data is included in your archive (for example Finder Info, Macintosh Resource Fork....). The information will be added to the archive in the same format as used by the Archive utility on Mac OS X.
- Variable name for added files: here you can specify the name of the variable to save the added files in. You can leave this field blank if you do not want to store the added files.

### Related concepts

Unzip files on page 159 With this node you can extract files from an archived file.

# Collaboration

### Wait for approval

With this node you can wait for the individual approvals.

### **Route After Approval**

With this node you can collect all the individual approvals and send the workable to a certain flow path depending on the assessment of the participants.

It has the following configuration options:

- **Policy**: here you can specify the routing. Options:
  - All Participants need to Accept: if you select this option, all participants need to accept to send the *workable* to the Approved output. The workable will be routed to Rejected as soon as *one* of the participants rejects.
  - All Participants need to access: if you select this option, all participants need to either accept or reject. The workable will be routed to the Approved output if all participants accepted. The workable will be routed to Rejected if one of the participants rejects (after *all* participants assessed the file).
  - Only One Participant needs to Accept or Reject: if you select this option, only one participant needs to accept or reject to send the workable to the corresponding output. This means that the assessment of the first participant will determine the routing.
  - All sub approvals need to be accepted (for combined approval): if you select this option, all sub approvals need to be accepted to send the workable to the Approved output. The workable will be routed to Rejected as soon as one of the sub approvals was rejected.
  - Last sub approval needs to be accepted (for combined approval): if you select this option, the last (started) sub approval need to be accepted to send the workable to the Approved output. The workable will be routed to Rejected if the last sub approval was rejected.
- Enable Timed-Out Output: if you select this option, the workable is sent to a Timed Out output if the final assessment is based on one or more timed-out workables.

### Set Approval Assessment

With this node you can set an approval assessment directly from within the workflow.

This node should be used on a *workable* that handles a participant approval or a sub-approval, not on the master approval workable.

It has the following configuration options:

- Activity: here you can select the activity. Options:
  - (Participant) Assess: with this option you set the participants assessment.
  - Cancel Approval: with this option you can cancel the current approval.
  - Force Approval: with this option you can force the current approval, ignoring the participants' decision.
- Assessment: here you can specify the value that will be assigned. Options:
  - Accept
  - Reject

### Start Approval

With this node you can start an approval cycle.

- File to approve: here you can define the file to add as reference to the approval records. You can leave this parameter empty if the approval is not related to a file. Select the pencil to open Expression Builder.
- **Combined**: if you select this checkbox, a *combined approval* is started. If you leave this checkbox unselected, a *regular approval* is started.

• Update Job: you only need to select this checkbox in case the approval is for a file in a Job. If you select this checkbox, the approval state of the file in the Job is updated by this approval.

=,

Note: You need to have a selected job when you select this checkbox.

- Enable Comment: if you select this checkbox, the participants are allowed to enter a comment to explain their assessment.
- Allow Conditional Accept: if you select this checkbox, (each) participant is allowed to *conditionally* accept. This means that the participant accepts the file, but only if the info in the comment is honored (for example if the participant specifies that only a word needs to be changed).
- Allow Delegating: if you select this checkbox, participants are allowed to delegate their assessment to other persons.
- On Restart: hold until file is updated: if you select this checkbox, the restart of an approval is held until a file has been updated.

#### Example

Suppose you want to restart an approval after a certain file was rejected, but you want to wait until the file is overwritten by the corrected file. In that case you can connect the output of the **Rejected** output of **Route After Approval** to the **Restart** input of this node and select this checkbox. The restart will only be triggered when the file has been overwritten. When you leave this checkbox unselected, the approval is restarted as soon as the *workable* arrives in this node.

• Variable name: here you can specify the name of the variable to store the Approval ID and Workable IDs in. If you want to set it to an empty string, this information is not needed.

### **Participants**

- User Names: here you can specify the user names of the users that need to participate in this approval cycle. See *Regular approvals* and *Combined approvals* for more information.
- E-mails: here you can specify the e-mail addresses of the users that need to participate in this approval cycle. See *Regular approvals* and *Combined approvals* for more information.
- Attributes: here you can specify the attributes of the users that need to participate in this approval cycle. See *Regular approvals* and *Combined approvals* for more information.

#### **Informative Participants**

- User Names: here you can specify the user names of the users that can participate in this approval cycle, but only on an informative level. They can add notes and accept/reject a file, but their assessment does not change the final assessment of the approval. See *Regular approvals* and *Combined approvals* for more information.
- E-mails: here you can specify the e-mail addresses of the users that can participate in this approval cycle, but only on an informative level. They can add notes and accept/reject a file, but their assessment does not change the final assessment of the approval. See *Regular approvals* and *Combined approvals* for more information.
- Attributes: here you can specify the attributes of the users that can participate in this approval cycle, but only on an informative level. They can add notes and accept/reject a file, but their assessment does not change the final assessment of the approval. See *Regular approvals* and *Combined approvals* for more information.

### **Team Leaders**

If you add one or more team leaders to an approval, these team leaders can manage their approvals (for example, add or remove participants, force approvals...).

• User Names: here you can specify the user names of the users that can administer this approval cycle. A team leader does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.

- E-mails: here you can specify the e-mail addresses of the users that can administer this approval cycle. A team leader does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.
- Attributes: here you can specify the attributes of the users that can administer this approval cycle. A team leader does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.

### Observers

If you add one or more observers to an approval, these observers are allowed to see the approval results, but cannot participate in the approval itself. They will also get the approval status from PROOFSCOPE.

- User Names: here you can specify the user names of the users that can follow this approval cycle. An observer does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.
- E-mails: here you can specify the user e-mail addresses of the users that can follow this approval cycle. An observer does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.
- Attributes: here you can specify the attributes of the users that can follow this approval cycle. An observer does not need to participate in the approval itself. See *Regular approvals* and *Combined approvals* for more information.
- **Display Form**: if you select this checkbox, the Approval Form will be displayed in the Sidebar in PROOFSCOPE when approving the file.
- Approval Form: here you can configure the parameters for the custom form that is displayed in PROOFSCOPE. This form fields needs to be filled in by the user in PROOFSCOPE. Select the pencil to open Expression Builder.

See Form Builder on page 260 for more information.

• Initial Form Data: here you can define the variable that contains the initial form data. You can define the content of the variable by a script in the Script node.

#### Example

You have an approval form that contains two fields:

- A text field with Label Customer and an id cusid.
- A text field with Label Job and an id jobid.

You have the following script in the Script node:

```
var values = {
    job: 'Job123',
    cusid: 'CustomerA'
};
setResultVariables({
    InitialValues: values
});
```

This means that the variable in the field **Initial Form Data** needs to be **InitialValues**. Consequently, the initial values of the form fields in PROOFSCOPE will be as follows:

- Customer CustomerA.
- Job CustomerA.
- **Approval Timeout**: here you can specify how long a user has to review the file. The approval state will be set to the **Time Out Assessment** if the user did not review the file within the specified period.

- **Approval Timeout Assessment**: here you can specify the value that will be assigned when the user did not review within the specified period.
- **Approval Reminder Interval**: here you can specify the time in minutes between two approval reminders. Use 0 when you do not want to send any reminders.
- Use Asset Notes: if you select this checkbox, the standard asset notes are used for this approval.
- Notes ID: here you can assign a notes ID. This parameter is only displayed if you leave the checkbox Use Asset Notes unselected. This ID will be used to create and/or to look up a specific notes record. Select the pencil to open Expression Builder.

## Assets

### Get Meta Data

With this node you can get the metadata from a file and save the info in a variable.

It has the following configuration options:

- File to get meta data from: here you can specify the file to get meta data from. Select the pencil to open Expression Builder.
- Variable name: here you can specify the name of the variable to store the meta data in. The default name is meta\_data.
- **Support Multiple Files**: Normally, the data from the file will be assigned directly to the specified variable. If you select this checkbox, data from multiple files is supported. In this case, an array will be assigned to the specified variable, and each entry will contain the info from one file. If you leave this checkbox unselected, the data from only one file is supported, because the info from the second file will overwrite the info from the first file.
- Only from file: with this checkbox you can specify whether the meta data can come from the asset info, or only directly from the file. This is sometimes important if you have custom meta data extractors. When you select this checkbox, the meta data can only come from the standard meta data extractors that will process the specified file. When you leave this checkbox unselected, the meta data can come from the asset (which might be updated/ changed when compared to the info obtained from the file), but on the other hand meta data might be available which cannot be obtained by the standard meta data extractors.
- **Include XMP data**: if you select this checkbox, the native XMP data block is included. A lot of the information in the XML block is copied to other parts of the metadata, so access to native XMP is not needed in most cases.
- Include information about embedded images and forms: if you select this checkbox, information about embedded images and forms is included in the returned meta data.

#### Add tag

With this node you can add a tag to a file.

It has the following configuration options:

- File to add a tag to: here you can specify the file to add a tag to. Select the pencil to open Expression Builder.
- **Name**: here you can specify the name of the tag to add. If a tag with that name is already assigned to the specified file, nothing will happen. Select the pencil to open Expression Builder.

### Get tags

With this node you can get the tags from a file and save the info in a variable.

- File to get tags from: here you can specify the file to get the tags from. Select the pencil to open Expression Builder.
- Variable Name: here you can specify the name of the variable to store the tags in. The default is tags.

### Remove tag

With this node you can remove a tag from a file.

It has the following configuration options:

- File to remove tag from: here you can specify the file to remove the tag from. Select the pencil to open Expression Builder.
- **Name**: here you can specify the name of the tag to remove. If there is no tag with that name assigned to the specified file, nothing will happen. Select the pencil to open Expression Builder.

#### Set Meta Data

With this node you can overwrite the meta data from a file.

It has the following configuration options:

- File to set meta data on: here you can specify the file to set the meta data on. Select the pencil to open Expression Builder.
- Meta data: here you can specify which meta data should be stored. Select the pencil to open Expression Builder.

## PROOFSCOPE

### Prepare PROOFSCOPE

With this node you can prepare a file to be viewed in PROOFSCOPE.

It has the following configuration options:

- Mode: here you can select the mode. Options:
  - From Files: if you select this option, the file(s) specified in the File to prepare parameter will all be prepared for viewing.
  - From Proofscope URL: if you select this option, the view that is referenced by a previously generated PROOFSCOPE URL (specified in the File to prepare parameter) will be looked up, and all assets referenced in that view will be prepared for viewing.
  - From View: if you select this option, the JSON as specified by the File to prepare parameter will be used to find assets, and all assets referenced in that view will be prepared for viewing.
- File to prepare: here you can specify which file to prepare. Select the pencil to open Expression Builder.
- Generate Text Layer: if you select this checkbox, a text layer for PDF files will be generated.
- **Render**: if you select this checkbox, the file will be rendered. The rendering will only take place if tiles do not exist yet.
  - **Render Always**: if you select this checkbox, the file will always be rendered. If the tiles already exist, they will be recreated.
  - Render Anti Aliassed: if you select this checkbox, anti-aliasing is applied during rendering.
  - **3D Render Type**: here you can select the render type from the drop-down list. Options:
    - Standard: if you select this option, standard rendering is used.
    - Ray Tracing: if you select this option, ray-tracing for rendering is used.
- Enable Extra 'no update' output: if you select this checkbox, an extra output connector will be created. The workable will follow this connector in case the node did not update the asset(s).

### Generate PROOFSCOPE URL

With this node you can create a URL that can be used to view one or more files in PROOFSCOPE.

- File to reference: here you can specify the file to create a URL for. Select the pencil to open Expression Builder.
- **Base Proofscope URL**: here you can specify the *base* URL where PROOFSCOPE should be accessed when a user opens the generated URL.
- Variable name: here you can specify the name of the variable to store the URL in. The default name is proofscope\_url.

- URL type: in this drop-down list you can select the type of URL to create. Options:
  - File: if you select this option, the generated URL will open PROOFSCOPE showing the specified file.
  - **Combined Files**: if you select this option, the generated URL will open PROOFSCOPE showing the specified files combined on top of each other.
  - File List: if you select this option, the generated URL will open PROOFSCOPE showing the specified files as if they were one multipage document.
  - File Difference: if you select this option, the generated URL will open PROOFSCOPE showing the specified file and *difference* file in difference mode.
  - Folder As Versions: if you select this option, the generated URL will open PROOFSCOPE showing all the files in a folder. The files are sorted by file name and filtered on mime type **application/pdf**, the last name being the current version. The files in the folder are listed as different versions of the same graphic.
  - Versions: if you select this option, the generated URL will open PROOFSCOPE showing all the specified files as different versions of the same graphic. The first file in the list is the current version.
  - 3D: if you select this option, the generated URL will open PROOFSCOPE showing the 3D file.
  - Artwork & 3D: the generated URL will open PROOFSCOPE showing a view where you can switch between the artwork file and the 3D file.
- **Difference File**: here you can select the file to compare with.<sup>3</sup>
- Version Sorting: here you can select the version sorting. Options:
  - Sorted Last file is current version: if you select this option, the files will be sorted in the following way:
    - A sequence of digits is handled as a single number.
    - The last file in the list is the current version, even if there is a file without a number.
  - Sorted File without number is current version: if you select this option, the files will be sorted in the following way:
    - A sequence of digits is handled as a single number.
    - The common part of (all) files is searched for, and the file that has no suffix is the current version.
  - Sorted File without number is first version: if you select this option, the files will be sorted in the following way:
    - A sequence of digits is handled as a single number.
    - The common part of (all) files is searched for, and the file that has no suffix is the first version.

4

- **Maximum nr pages**: here you can limit the number of pages that is displayed by PROOFSCOPR for multipage files. If you set this parameter to **0**, all pages of a multipage file are displayed. If you set it to another number, this will be the maximum number of pages that is displayed. This maximum value will be ignored if the file itself contains less pages than this maximum.
- Enable Approval: if you select this checkbox, the generated URL is intended to be used for approval. This also requires that the *workable* already contains an Approval ID, added by a Start Approval node. If you leave this checkbox unselected, the generated URL can be used for viewing only.
- Show All Participants Assessment: if you select this checkbox, PROOFSCOPE displays an extra tab with the approval assessment details.
- Enable Guest access: if you select this checkbox, the URL can be used by guests who are not logged in on the system. In this case, a valid name must be specified in the field Email address. If you leave this checkbox unselected, the URL can only be used by users who are logged in.

<sup>&</sup>lt;sup>3</sup> This parameter is only available if you selected **File Difference** as URL type.

<sup>&</sup>lt;sup>4</sup> This parameter is only available if you selected Folder As Versions as URL type.

- Email address: here you can specify the e-mail address of the invited guest. This field must have a valid e-mail address when the checkboxes Enable approval and/or Enable Guest access are selected.
  - If neither **Enable approval** nor **Enable Guest access** are selected but the field does have a value, the value is used to do a check if the specified user is able to view the file. If for example the scope of the asset and the user do not match, no URL will be generated.
  - If neither **Enable approval** nor **Enable Guest access** are selected and the field does not have a value, no check is done.
- **Require Login (for existing users)**: if you select this checkbox, existing users need to login when opening the generated URL.

### **Render Configuration**

- **Invert Colorants**: if you select this checkbox, all colorants will be displayed inverted and the viewer will need to invert all colorants. If you leave this checkbox unselected, all colorants will be displayed in a normal way.
- **Render Single Colorant As Black**: if you select this checkbox, a single colorant will be displayed in black&white and the viewer needs to switch to black&white view. If you leave this checkbox unselected, a single colorant will be displayed in the correct color. This option is only effective if just one colorant is displayed.

### **Proofscope Configuration**

• Intro page URL: here you can specify an HTML page with extra information and/or explanation on PROOFSCOPE or your approval cycles. The content of the HTML is free to choose and it is visible as an overlay when PROOFSCOPE is opened.

The page needs to be hosted on a website (for example http://myserver.com/path/to/intro.html) or needs to be accessible through a File store (for example cloudflow://PP\_FILE\_STORE/path/to/ intro.html).

**Note:** If you add the HTML file in a File store, make sure that the folder is accessible for everyone who logs in to CLOUDFLOW.

### Example



• Allow Operator to save configuration changes: if you select this checkbox, operators are allowed to save configuration changes. This can be used to create a URL, then let a first operator further configure the view (for example, set the channel mapping correctly or align 2 difference files), and only then send the URL to other users that will then open PROOFSCOPE with the fully configured view.

- Show Notes: if you select this checkbox, the user will be able to view, change and add notes, depending on the selected checkboxes below.
  - Can delete Notes: if you select this checkbox, users can delete their own notes. The owner of a note is the owner of the first comment on that note.
  - Can only view notes: if you select this checkbox, users can only view notes.
  - Show notes history filter: if you select this checkbox, PROOFSCOPE shows a notes history filter.
  - Show notes from older versions when opening Proofscope: if you select this checkbox, PROOFSCOPE shows notes from older versions.
  - Make Notes View Only After Assessment: if you select this checkbox, notes will become view-only as soon as a participant assessed a file.

**Note:** This checkbox is only available if **Enable Approval** is selected.

• **Require note on reject**: if you select this checkbox, participants are required to add at least one note when they reject a file.

**Note:** This checkbox is only available if **Enable Approval** is selected.

- **Show File Info Details**: if you select this checkbox, the **Separations** on page 305 will be visible in the PROOFSCOPE Sidebar. If you deselect the checkbox, the File Info Details will not be visible.
  - Show page boxes: if you select this checkbox, the page boxes will be visible in the File Info Details in PROOFSCOPE.
  - Show page size info: if you select this checkbox, the page size will be visible in the File Info Details in PROOFSCOPE.
  - Box to use for page size: here you can define the box to use for the page size.
  - Show Distortion Information: if you select this checkbox, the distortion information that is in the file will be visible in the File Info Details in PROOFSCOPE.

Note: This option is only available if you have selected the checkboxes Enable generation of previews controlled by workflow and Enable Distortion in Proofscope in SETTINGS > SETTNIGS > QUANTUM.

- Show Measure Density Tool: if you select this checkbox, the user will have access to the Densitometer tool. If you leave this checkbox unselected, the user will not have access to the tool.
- Show Measure Tool: if you select this checkbox, the user will have access to the Measure Tool in case it is available for the file being viewed.
- Show Rotate Buttons: if you select this checkbox, the user will have access to the Rotate Buttons in case it is available for the file being viewed.
- Show Mirror Tool: if you select this checkbox, the user will have access to the Mirror Tool in case it is available for the file being viewed.
- QA Tools
  - Show Measure Halftone Tool: if you select this checkbox, the user will have access to the Measure Halftones tool in case it is available for the file being viewed.
  - Show Read Barcode Tool: if you select this checkbox, the user will have access to the Detect Barcodes tool in case it is available for the file being viewed.
  - Show Minimum Dot Tools: if you select this checkbox, the user will have access to the Minimum Dot Tools in the Sidebar.
  - Show Maximum Dot Tools: if you select this checkbox, the user will have access to the Maximum Dot Tools in the Sidebar.
  - Show Total Area Coverage Tools: if you select this checkbox, the user will have access to the Total Area Coverage Tools in the Sidebar.
  - Show Invert Separation Order Checkbox: if you select this checkbox, the user will have access to the Invert Separation Order Checkbox in the Sidebar.
- Show Note-From-Text Tool: if you select this checkbox, the user will have access to the Read note From Text tool in case it is available for the file being viewed.

- Show Preflight Notes: if you select this checkbox, (optional) preflight notes associated with assets are shown in the PROOFSCOPE sidebar.
- Show Download Low Resolution Document: if you select this checkbox, the user will be able to download a low resolution version of the document being viewed.
- Show Download Notes Report: if you select this checkbox, the user will be able to download a report containing a summary of the notes, the approval status... of the document being viewed.

### Save PROOFSCOPE View To File

With this node you can rasterize a file and save it to a file.

It has the following configuration options:

- Mode: here you can select how the view configuration will be fetched. Options:
  - From URL: if you select this option, the view referenced by a (previously generated) PROOFSCOPE URL (as specified in the view parameter) will be looked up, and that view will be rendered and saved.
  - From View: if you select this option, the JSON as specified by the view parameter will be used to construct a PROOFSCOPE view, and that view will be rendered and saved.
- View: here you can specify the view to render, or the URL that contains the view reference.
- Output file: here you can specify the name and path of the file that needs to be generated.
- **Resolution**: here you can specify the resolution that will be used to rasterize the data. The generated raster data is 8 bits per pixel RGB.
- Render Anti Aliassed: if you select this checkbox, the file will be rendered with anti-aliasing.
- Lossless compression: if you do not select this checkbox, the image data in the generated PDF file is compressed using JPEG, and may contain compression artifacts. This might be a problem if the saved file needs to be printed. If you select this checkbox, any compression artifact is removed.

**Note:** Compression artifacts in the initial data might still be visible in the output.

• **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### Update PROOFSCOPE Configuration

With this node you can update an existing PROOFSCOPE view configuration (for example hiding the colorants in a later stage of approval).

It has the following configuration options:

• View: here you can specify the URL that contains the view reference.

### **Render Configuration**

- **Invert Colorants**: if you select this checkbox, all colorants will be displayed inverted and the viewer will need to invert all colorants. If you leave this checkbox unselected, all colorants will be displayed in a normal way.
- **Render Single Colorant As Black**: if you select this checkbox, a single colorant will be displayed in black&white and the viewer needs to switch to black&white view. If you leave this checkbox unselected, a single colorant will be displayed in the correct color. This option is only effective if just one colorant is displayed.

### **Proofscope Configuration**

• Intro page URL: here you can specify an HTML page with extra information and/or explanation on PROOFSCOPE or your approval cycles. The content of the HTML is free to choose and it is visible as an overlay when PROOFSCOPE is opened.

The page needs to be hosted on a website (for example http://myserver.com/path/to/intro.html) or needs to be accessible through a File store (for example cloudflow://PP\_FILE\_STORE/path/to/ intro.html).

**Note:** If you add the HTML file in a File store, make sure that the folder is accessible for everyone who logs in to CLOUDFLOW.

### Example



• Allow Operator to save configuration changes: if you select this checkbox, operators are allowed to save configuration changes. This can be used to create a URL, then let a first operator further configure the view (for example, set the channel mapping correctly or align 2 difference files), and only then send the URL to other users that will then open PROOFSCOPE with the fully configured view.

- Show Notes: if you select this checkbox, the user will be able to view, change and add notes, depending on the selected checkboxes below.
  - Can delete Notes: if you select this checkbox, users can delete their own notes. The owner of a note is the owner of the first comment on that note.
  - Can only view notes: if you select this checkbox, users can only view notes.
  - Show notes history filter: if you select this checkbox, PROOFSCOPE shows a notes history filter.
  - Show notes from older versions when opening Proofscope: if you select this checkbox, PROOFSCOPE shows notes from older versions.
  - Make Notes View Only After Assessment: if you select this checkbox, notes will become view-only as soon as a participant assessed a file.

**Note:** This checkbox is only available if **Enable Approval** is selected.

• **Require note on reject**: if you select this checkbox, participants are required to add at least one note when they reject a file.

**Note:** This checkbox is only available if **Enable Approval** is selected.

- **Show File Info Details**: if you select this checkbox, the **Separations** on page 305 will be visible in the PROOFSCOPE Sidebar. If you deselect the checkbox, the File Info Details will not be visible.
  - Show page boxes: if you select this checkbox, the page boxes will be visible in the File Info Details in PROOFSCOPE.
  - Show page size info: if you select this checkbox, the page size will be visible in the File Info Details in PROOFSCOPE.
  - Box to use for page size: here you can define the box to use for the page size.
  - Show Distortion Information: if you select this checkbox, the distortion information that is in the file will be visible in the File Info Details in PROOFSCOPE.

Note: This option is only available if you have selected the checkboxes Enable generation of previews controlled by workflow and Enable Distortion in Proofscope in SETTINGS > SETTNIGS > QUANTUM.

- Show Measure Density Tool: if you select this checkbox, the user will have access to the Densitometer tool. If you leave this checkbox unselected, the user will not have access to the tool.
- Show Measure Tool: if you select this checkbox, the user will have access to the Measure Tool in case it is available for the file being viewed.
- Show Rotate Buttons: if you select this checkbox, the user will have access to the Rotate Buttons in case it is available for the file being viewed.
- Show Mirror Tool: if you select this checkbox, the user will have access to the Mirror Tool in case it is available for the file being viewed.
- QA Tools
  - Show Measure Halftone Tool: if you select this checkbox, the user will have access to the Measure Halftones tool in case it is available for the file being viewed.
  - Show Read Barcode Tool: if you select this checkbox, the user will have access to the Detect Barcodes tool in case it is available for the file being viewed.
  - Show Minimum Dot Tools: if you select this checkbox, the user will have access to the Minimum Dot Tools in the Sidebar.
  - Show Maximum Dot Tools: if you select this checkbox, the user will have access to the Maximum Dot Tools in the Sidebar.
  - Show Total Area Coverage Tools: if you select this checkbox, the user will have access to the Total Area Coverage Tools in the Sidebar.
  - Show Invert Separation Order Checkbox: if you select this checkbox, the user will have access to the Invert Separation Order Checkbox in the Sidebar.
- Show Note-From-Text Tool: if you select this checkbox, the user will have access to the Read note From Text tool in case it is available for the file being viewed.

- Show Preflight Notes: if you select this checkbox, (optional) preflight notes associated with assets are shown in the PROOFSCOPE sidebar.
- Show Download Low Resolution Document: if you select this checkbox, the user will be able to download a low resolution version of the document being viewed.
- Show Download Notes Report: if you select this checkbox, the user will be able to download a report containing a summary of the notes, the approval status... of the document being viewed.

# PACKZflow

With the PACKZ flow nodes you can build workflows to automate prepress production.

**Note:** If you're working on a Windows server, disable the Deduplication Services functionality. This is not supported and can cause the PACKZflow nodes to error.

## **AI Roundtrip**

With this node you can define how an Illustrator (.AI) file should be opened.

It has the following configuration options:

- Input File: here you can specify the input file. Select the pencil to open Expression Builder.
- File Open Mode: here you can define how the Illustrator file should be opened. Options:
  - **Prefer PDF**: if you select this option, the PDF supported part is read, ignoring everything that is written in the Illustrator part such as text blocks, image links and layers.
  - **Prefer Illustrator**: if you select this option, the Illustrator part will be imported, translating the Illustrator part into editable PDF language. This means that:
    - The image links will be retrieved.
    - The layers will be kept.
    - The text blocks will be kept.
  - **Illustrator Only**: if you select this option, only the Illustrator part of the PDF file is accepted. If this is not available, an error will occur.
- File Save Mode: here you can define how the file should be saved. Options:
  - Save as PDF: if you select this option, the file is saved as regular PDF file.
    - **Note:** If you want to process the file in a further step in the workflow, you need to use this option. If you need an Illustrator file as a result, you need to use another AI Roundtrip node at the end of your workflow.
  - Save as Illustrator: if you select this option, the input file is exported as an Illustrator file.

**Note:** It is not possible to process native Illustrator files in further steps in the workflow. Consequently, this option should always be the last step in the workflow.

• **Image Lookup Folder**: here you can define the folder path where the images are located. The path can be absolute or relative.

#### Example

=

If you define the relative path . . / Images, the node will look for images in the **Images** folder that is in a parent folder of the imported AI document.

You can also enter a variable containing a list of folders (in which case the variable must be a list of strings).

- Missing Font Handling: here you can define what needs to happen when a font is missing. Options:
  - Error When Missing: if you select this option, the workflow will error.
  - Substitute Font: if you select this option, the font will be substituted by the font specified in Default Font.
  - Default Font: here you can specify which font is used when a missing font is substituted.
- Output AI file: here you can define the output path of the Illustrator file.

### Analyze

With this node you can perform a check on the content PDF document, based on a list of criteria.

It has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

You can change the severity of the check results by selecting one of the icons next to the checkboxes.

- 🕖 will result in information.
- 🔺 will result in a warning.
- 😑 will result in an error.
- **High resolution images**: if you select this checkbox, the document will be analyzed for images with a resolution equal to or higher than the resolution specified in the input field. The search will also include images used as a soft mask.
  - **Downsample**: if you activate this button, high resolution images will be downsampled. Images within a station will also be downsampled.

**Note:** Downsampling images can significantly reduce the size of a PDF as it reduces the number of pixels in an image. However, this also means the quality of the image will decrease.

- **High Resolution Bitmaps**: if you select this checkbox, the document will be analyzed against bitmap images with a resolution equal to or higher than the resolution specified in the input field.
  - **Downsample**: if you activate this button, high resolution bitmap images will be downsampled. Bitmap images within a station will also be downsampled.
- Low Resolution Images: if you select this checkbox, the document will be analyzed against images with a resolution equal to or lower than the resolution specified in the input field.
- Low Resolution Bitmaps: if you select this checkbox, the document will be analyzed against bitmaps with a resolution equal to or lower than the resolution specified in the input field.
- **Bitmaps**: if you select this checkbox, the document will be analyzed against bitmap images (images with only two different pixel colors).
- **Out of Sync Images**: if you select this checkbox, the system will check if the images within the document are still up-to-date with the original images. When there are stations within the document, the content of the station will also be checked on outdated image information.
  - **Reload**: if you activate this button, active images in the document will update accordingly to the images on the disk.
- **Missing Image Files**: if you select this checkbox, the document will be analyzed against images that have a file path which points to a file that doesn't exist.
- Pattern Paints: if you select this checkbox, the document will be analyzed against objects with a pattern as paint.
  - Vectorize: if you activate this button, patterns will be converted to paths.
- **Detect Barcodes**: if you select this checkbox, the document will be analyzed against barcodes that can be recognized by the **Barcode Recognize** function in **PACKZ**.
  - **Recognize**: if you select this checkbox, barcodes that are created in another application will be recognized and updated if needed.
- **Mixed Paint Barcodes**: if you select this checkbox, the system will perform a barcode detection (not a recognition) and check the detected barcodes for mixed paint objects. A detected barcode is added to the collection if at least one contained object has a mixed paint. Also barcodes in RGB or Lab will be marked in the Mixed Paint Barcodes collection.

- Illustrator Blends: if you select this checkbox, the document will be analyzed against *Illustrator blends*.
  - To Image: if you activate this button, Illustrator blends will be converted to images.
- **Illustrator Deskpack 0.2% Colors**: if you select this checkbox, the document will be analyzed against 0.2% ink components created by Deskpack to simulate the overprint that is not supported by Illustrator.
  - To 0% and DeviceN: if you activate this button, the 0.2% ink components created by Deskpack will be put back to 0%.
- **Redundant Clips**: if you select this checkbox, the document will be analyzed against clipping objects that do no clip anything.
  - **Remove**: if you activate this button, all redundant clips will be removed.
- **Out of Sync Stations**: if you select this checkbox, the document will be analyzed against station content that is no longer up-to-date with the original content.
  - **Reload**: if you activate this button, active stations in the document will be updated according to the stations on the disk.
- Distortion: if you select this checkbox, the document will be analyzed against the use of a distortion.
- Printable Notes: if you select this checkbox, the document will be analyzed against printable notes. Options:
  - Collect Only: if you select this option, the printable notes will be listed in a collection.
  - **Remove**: if you select this option, all printable notes will be removed.
  - Remove Unsupported: if you select this option, only unsupported printable notes will be removed.
  - Clear Printable Flag: if you select this option, the printable flag will be removed from the notes.
- **PDF Layers**: if you select this checkbox, the document will be analyzed against PDF Layers that are hidden (**Hidden**) or hidden but printable (**Hidden but Printable**). Options:
  - Collect Only: if you select this option, hidden or hidden but printable layers will be listed in a collection.
  - Make Visible: if you select this option, hidden or hidden but printable layers will be made visible.
  - Make non-Printable: if you select this option, hidden or hidden but printable layers will be made non-printable.
- None Embedded Fonts: if you select this checkbox, the document will be analyzed against non-embedded fonts.
- Missing Font Symbols: if you select this checkbox, the document will be analyzed against missing font outlines.
- Subset Fonts: if you select this checkbox, the document will be analyzed against subset fonts.
- Text Objects: if you select this checkbox, the document will be analyzed against all live text.
  - Vectorize: if you activate this button, all live text will be converted to outlines.
- **Text In Page Box Margin**: if you select this checkbox, the document will be analyzed against text that is too close to a certain page box border. You can select the page box from the drop-down menu and you can set the margin value in the corresponding input field.
  - Page Box:: here you can select the page box from the drop-down list.
  - Margin:: here you can define the margin value.
- **Mixed Paint Text**: if you select this checkbox, the document will be analyzed against text that has more than one separation color, or text in RGB or Lab. Text part of a recognized barcode or barcode created in PACKZ will be skipped. Within a text group, all consecutive characters having the same matching criterion (mixed fill paint, mixed stroke paint or both) will be grouped into a single collection item.

- Font Size: if you select this checkbox, the document will be analyzed against text that has a certain font size. Options:
  - At least: if you select this option, the check searches for text that is bigger than or equal to the size entered in the input box.
  - Exactly: if you select this option, the check searches for text that is exactly the size entered in the input box.
  - At most: if you select this checkbox, the check searches for text that is smaller than or equal to the size entered in the input box.
  - Number of separations at least: here you can define the number of separations for both the fill and stroke of the text. If no separation colors are used in the text and the Number of separations at least parameter is set to 1, it will still be added to the collection, but in stead of a number a question mark will be displayed.
- EU 1169/2011 Violations (1,2 mm): if you select this checkbox, the document will be analyzed against (live) text that does not meet the EU 1169/2011 regulation concerning the minimum font size. This regulation specifies that the font's minimum height of the lower case x character should be at least 1.2 mm.
- EU 1169/2011 Violations (small packages: 0,9mm): if you select this checkbox, the document will be analyzed against (live) text that does not meet the EU 1169/2011 regulation concerning the minimum font size on packages for which the largest surface has an area of less than 80 cm2. This regulation specifies that the font's minimum height of the lower case x character should be at least 0.9 mm.
- **RGB Objects**: if you select this checkbox, the document is analyzed against objects in an RGB color space.
  - Convert To CMYK: if you activate this button, active RGB objects will be converted to CMYK.
- Lab Objects: if you select this checkbox, the document is analyzed against objects in objects in Lab.
  - Convert To CMYK: if you activate this button, active Lab objects will be converted to CMYK.
- **Gray Objects**: if you select this checkbox, the document will be analyzed against objects in a Gray color space (except when they are used in soft mask).
  - **Make Black**: if you activate this button, active objects in color space Gray (except when they're used in soft mask) will be converted to Black.
- **Registration Objects**: if you select this checkbox, the document will be analyzed against objects in registration paint. 0% registration objects will be ignored.
  - **Percentage**: here you can define the registration percentage value.
  - **Convert to Separated**: if you activate this button, the selected registration objects will get a Separated modus paint.
- ICC Profiles: if you select this checkbox, the document will be analyzed against objects containing an ICC Profile. From the drop-down list you can select an option to only search for ICC profiles in images, in CMYK images or in all kind of objects.
  - **Remove**: if you activate this button, objects or images containing a profile will be removed.
- **Different Profiles with Same Name**: if you select this checkbox, the document will be analyzed against profiles with the same name (case insensitive) and binary equality.
  - **Resolve**: if you activate this button, these profiles will be replaced with the first one that was encountered in the document.
- Knockout Black Objects: if you select this checkbox, the document will be analyzed against all objects that contain only 100% of the Black separation. This can be strokes and/or fills of any object and text.
  - Make Overprinting: if you activate this button, active opaque inks that are in knockout will be put in overprint.

- Knockout Black Text: if you select this checkbox, the document will be analyzed against text that has a fill paint of 100% Black.
  - Max. Size: if you activate this button, you can indicate the maximum size of the black text.
  - Make Overprinting: if you activate this button, black text will be put in overprint.
- Knockout Black Strokes: if you select this checkbox, the document will be analyzed against black strokes in knockout, except strokes on text. For text, you can use the Knockout Black Text check.
  - Max. Width: here you can define a maximum stroke width.
  - Make Overprinting: if you activate this button, black strokes will be put in overprint.
- Knockout Black Rectangles: if you select this checkbox, the document will be analyzed against rectangles that have a black fill in knockout.
  - Max. Width: here you can define a maximum width, which is the smallest length of the rectangle.
  - Make Overprinting: if you activate this button, black rectangles will be put in overprint.
- **Overprints**: if you select this checkbox, the document will be analyzed against objects in overprint.
  - Analyze fill: if you select this button, the check will search for objects with a fill in overprint.
  - Analyze stroke: if you select this button, the check will search for objects with a stroke in overprint.
- **Overprint Mode 1**: if you select this checkbox, the document will be analyzed against objects containing an Overprint mode 1 construct.
  - **Resolve**: if you activate this button, these constructions will be converted to real overprints.
- **Technical inks in knockout**: if you select this checkbox, the document will be analyzed against objects containing one or more technical inks in knockout.
  - Make Overprinting: if you activate this button, active technical inks that are in knockout will be put in overprint.
- Varnishes in knockout: if you select this checkbox, the document will be analyzed against objects containing one or more varnish inks in knockout.
  - Make Overprinting: if you activate this button, active varnish inks that are in knockout will be put in overprint.
- **Opaque inks in knockout**: if you select this checkbox, the document will be analyzed against objects containing one or more opaque inks. Objects matching this check will have an indication in the collection's item name informing you if the opaque ink in knockout was detected in the fill, stroke or both. Objects will not be detected in the following cases:
  - If the object (or object's group) has blend mode **Darken** or **Multiply**.
  - If the object's paint has at least one non-technical ink value set to zero.
  - **Make Overprinting**: if you activate this button, active opaque inks that are in knockout will be put in overprint.
- White Objects in Overprint: if you select this checkbox, the document will be analyzed against white text or white objects in overprint. White RGB and grayscale objects will be included. Lab color space is not supported by this function.
  - Analyze text objects: if you activate this button, the check will search for white text objects in overprint.
  - Analyze other objects: if you activate this button, the check will search for other white objects in overprint.
  - **Remove Overprint**: if you activate this button, the overprint will be removed and the objects will be put in white.
- **Mixed Paint Objects**: if you select this checkbox, the document will be analyzed against objects with mixed paint. Contrary to the **Mixed Paint Barcodes** criteria, it will not perform a barcode recognition prior to the check (if not recognized yet) and its individual components will be flagged.
- **Technical/Non-Technical Ink Mix**: if you select this checkbox, the document will be analyzed against objects containing both a mix of technical and non-technical ink.

- Non-Separable Blend Modes: if you select this checkbox, the document will be analyzed against objects containing a *non-separable blend mode*.
- **Invisible paints**: if you select this checkbox, the document will be analyzed against objects that have a paint in which all the separations are set to overprint. Options:
  - Collect Only: if you select this option, these objects are collected.
  - **Resolve**: if you select this option, these objects will not be included on the output as some RIPs struggle with these kinds of objects.
  - **Remove Redundant Objects**: if you select this option, redundant objects will be removed, but only if they are not marked as a trim path. In case the object is defined as trim path, the paint will be resolved instead.
- **Opaque Inks and Varnishes Ink Types**: if you select this checkbox, the document will be analyzed against opaque inks, varnishes or both. An ink is considered to be opaque as soon as the opacity is not zero. Options:
  - **Opaque Inks**: if you select this option, the check will search for opaque inks.
  - Varnishes: if you select this option, the check will search for varnishes.
  - **Opaque Inks and Varnishes**: if you select this option, the check will search for opaque inks and varnishes.
- **Combined Overprint and Transparency**: if you select this checkbox, the document will be analyzed against objects that have an overprint in both a fill or a stroke and a transparency.
- Blend Mode and Transparency: if you select this checkbox, the document will be analyzed against images containing an alpha channel and/or objects that have:
  - Any blend mode apart from Normal.
  - An opacity value lower than 100%.

This also includes group transparencies or blend modes applied on a group.

- **Object Separation Usage**: if you select this checkbox, the document will be analyzed against objects and/or text for which at least one separation has a percentage equal to or below the specified value. This check will only report flat paints with a gray or separated color space.
  - Analyze text objects: if you activate this button, the check will search for text objects.
  - Analyze other objects: if you activate this button, the check will search for other objects.
  - **Percentage at Most**: here you can specify the maximum percentage value of the separation(s).
- Unused Separations: if you select this checkbox, the document will be analyzed against unused separations.
  - **Remove**: if you activate this button, unused separations will be removed.
- Screened Objects: if you select this checkbox, the document will be analyzed against objects containing screening. The collection will list the matching items with the screening name.
  - **Remove Screening**: if you activate this button, the screening parameters from the objects containing screening will be removed.
- Screening in XMP: if you select this checkbox, the document will be analyzed against screening descriptions in the XMP. Regardless of the number of unscreened objects, only a single item will be reported in the collection. The item in the collection will display the screening information that was found.
  - Set As Default: if you activate this button, the screening found in the XMP will be applied to all the objects in the file.
- Stroke: if you select this checkbox, the document will be analyzed against specific strokes.
  - Separation Count: here you can define a number of separations. You can select At least, Exactly or At most from the drop-down menu.
  - Stroke Width: here you can define the stroke width. You can choose At least, Exactly or At most from the drop-down menu.
  - Fix: if you activate this button, the strokes will be set to the desired stroke width.

- **Open Paths**: if you select this checkbox, the document will be analyzed against open paths.
  - Close: if you activate this button, all open paths from the same object will be closed.
- **Suggested Text Replacement**: if you select this checkbox, the document will be analyzed against specific text parts, which you can replace or correct by other text. These specific text parts and its replacement text are defined in the **TextReplacements.json** file in the **Shared Application Data** folder.
  - **Replace**: if you activate this button, the selected text is replaced by the suggestion defined in the **TextReplacements.json** file.
- **Report to variable**: here you can define the name of the variable that will be created. The variable will contain information (type, location...) of the issues that are found in the document, based on the selected criteria.
- Save Preflight Report in Asset: if you select this checkbox, the result of the analyze node will be saved in the asset (in the database). This information can for example be used to show the preflight result in the PROOFSCOPE sidebar.
- **Inspect**: here you can limit the search to a particular page box. Options:
  - All Objects
  - Objects in Media Box
  - Objects in Crop Box
  - Objects in Bleed Box
  - Objects in Trim Box
  - Objects in Art Box
  - Objects in Trim Path

If you have selected an option different from **All Objects**, only objects that have a bounding box that overlaps the selected page box will be analyzed. If the page box does not exist, the analyze process falls back to the default, which is **All Objects**. If there is no **Trim Path** defined and **Inspect** is set to **Trim Path**, the **Trim Box** will be taken.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

#### Barcode

This node allows you to recognize different types of barcode placeholders created with any other application and update the type with a new code.

It has the following configuration options:

- Input File: here you can select the input file. Select the pencil to open Expression Builder.
- **Type**: here you can define the type of barcode to be recognized/updated in case a specific type needs to be recognized or updated.

#### Recognize

• **Recognize barcode**: if you select this checkbox, barcodes that are created in another application will be recognized and updated if needed.

- Count: here you can check if a minimum or specific number of barcodes is present in the design. Options:
  - No value: if you select this option, no check is done.
  - At Least: if you select this option, you can define a minimum number of barcodes that need to be present in the design. If there are less barcodes found in the design than what is specified in this field, an error will be generated.
  - Exactly: if you select this option, you can define a specific number of barcodes that needs to be present in the design. If there are more/less barcodes found in the design than what is specified in this field, an error will be generated.
  - **Note:** If you have selected a specific type of barcode from the **Type** list, only the barcodes of that type will be counted. If there is no type defined, all barcodes will be counted.

### Update

- Bar Width Reduction: here you can specify the *bar width reduction* that needs to be applied.
- Code: here you can specify the updated code that needs to be applied.
- Update Magnification: if you select this checkbox, the magnification will be updated.
- **Magnification**: here you can define the magnification. The magnification describes the bars and digits and ensures an optimal compatibility for the scanner.
- Update Quiet Zone: if you select this checkbox, the quiet zone will be updated.
- **Include Quiet Zone Indicator**: if you select this checkbox, a quiet zone indicator is included. This is used for scanners reading the digits instead of the bars. It is added to the right side of the barcode so that those scanners can properly read the code. A quiet zone is the blank margin on either side of a bar code that's used to tell the barcode reader where a barcode starts and stops. The purpose of a quiet zone is to prevent the reader from picking up information that does not pertain to the bar code that is being scanned. The blank margin will not send a scanning signal, hence the name **quiet**.
- Update Custom Height: if you select this checkbox, the custom height will be updated.
- Use Custom Height: if you select this checkbox, a custom height is used. Although a barcode has a predefined height, according to the magnification factor defined in the GS1 standard, you may want a different height.
- **Height**: here you can define the custom bar height, in preferred units. The height should be at least 25,9300 mm for a Magnification value of 100%.
- Update Clear Area: if you select this checkbox, the clear area will be updated.
- Create Clear Area: if you select this checkbox, a clear area will be created. This option adds a white rectangle to the barcode, knocking out the objects underneath it. The different icons with input fields underneath the checkbox represent the left / right / top / bottom margins. The margin specified is calculated from the borders of the barcode.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

#### **Color Conversion**

With this node you can do color conversion.

It has the following configuration options:

- Input File Test: here you can define the input file. Select the pencil to open Expression Builder.
- Selection only: if you select this checkbox, only the objects that were selected (for example from a previous node) will be taken into account for the color conversion.

### Profile Source

• **Color Profile**: here you can add the input ICC profiles. Select + to add an input color profile. Select - to delete an input profile.
- Exclude Separations: here you can specify one or multiple separations that are in the input file and that are described in the input ICC profile but that you want to exclude from the color conversion. For example, if you want to keep 100% black as 100% black, add black to the list of separations to exclude. Use a comma separated list if you want to define multiple separations.
- **Do not convert separation names**: here you can specify separation names in the input file that you do not want to convert. For example **Diecut** or **Varnish**. Use a comma separated list if you want to define multiple separations.

## **Destination Color Space**

- Output Profile: here you can define the destination ICC profile.
- Use profile from Device Link: if you select this checkbox, a DeviceLink profile will be used. A DeviceLink profile represents a complete color transformation from input to output, without the use of an independent profile connection space. It can only be used for a conversion between two specific device color spaces.
- Assign profile to artwork : if you select this checkbox, the profile will be assigned to the artwork.

# Options

**Rendering Intent**: here you can select the way color conversion from one color space to another will be handled. It will also determine also how the source colors are adjusted when they exceed the output gamut. Options:

- **Perceptual**: if you select this option, the colors will be rendered in a way that is natural to the human eye by scaling the entire input gamut to the output gamut while preserving a distinct relationship between the gamut colors. Perceptual is mostly used for images and the Japanese printing industry.
- **Relative Colorimetric**: if you select this option, the white point from the input gamut will be mapped to the white point of output gamut and renders all colors in relation to this mapping. Colors outside the gamut are mapped to the closest color of the output gamut. Relative Colorimetric is used in the European and North American printing industry.
- **Saturation**: if you select this option, the entire input gamut will be scaled to the output gamut while preserving or increasing the saturation of the colors in order to keep them vivid. This will influence changes in hue and lightness, which is unacceptable in the printing industry. Saturation will be used for business graphics like graphs or charts.
- Absolute Colorimetric: if you select this option, the colors will be displaced so that the source white point aligns with the destination white point, while Absolute Colorimetric preserves the white point and in gamut colors. Out of gamut colors will be clipped. This rendering intent is useful for previewing how paper color affects printed colors.

# **On Input Profile Mismatch**

In this section you can determine what happens when the input profile is missing or does not match the working color space. This can happen when opening a document or copying and pasting objects from one document to another containing a profile that is different from the current color space profile or containing no profile at all.

- Ignore object profile: if you select this checkbox, the object profile will be ignored.
- Allow profile mixing if you select this checkbox, multiple profiles will be combined for the conversion of the color.
- Allow Lab based conversion: if you select this checkbox, the Lab value of the color will be used for the conversion.

**Note:** This only works when an output profile is defined (no DeviceLink profiles).

• Allow default conversion: if you select this checkbox, the default conversion will be done.

## **Extended Gamut Conversion**

=

• **Remap Separation Names**: here you can define the list of separation names that have the Extended Gamut profile mapping attached.

**Note:** You need to install a color book with Extended Gamut Conversion definition.

• Error if no separations match: if you select this option, the workflow will error if the none of the separations listed could be converted to Extended Gamut.

## **Convert to Image**

With this node you can convert an object or a group of objects to an image equivalent.

It can be used to:

- Reduce the complexity of a layout, label or package.
- Clean up sliced background images in PDF documents created with an application that exports the proprietary file format to PDF

The node has the following configuration options:

- Input File: here you can select the input file. Select the pencil to open Expression Builder.
- **Convert Selection Only**: if you select this checkbox, only a selected group of objects will be converted to an image. If you leave this checkbox unselected, any object in the file will be converted to an image.
- **Guarantee Visual Correctness**: if you select this checkbox, the knockout objects lying on top of the selected objects will be taken into account during the conversion, which creates a knockout on the objects converted to image. If you leave this checkbox unselected, knockout objects lying on top of the selected objects will be ignored while converting the selected objects to image.

### General

- **Resolution**: here you can select the resolution of the image that will be generated.
- **Oversampling**: here you can select the *oversampling* ratio. For example, if the resolution is set to 300 ppi and the oversampling ratio to 2, the images will be converted at 600 ppi and downsampled to 300 ppi.

### Shadings

- Noise: here you can define how much noise should be applied to the shadings when converting to image to prevent them from banding on the image setter or proofer. However, noise may cause screen defects when using hybrid screens in which conventional and frequency-modulated screens are combined. Therefore noise can be applied in certain ranges, between certain percentages:
  - Apply from 0% to ...%
  - And from ...% to 100%
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### Dimensions

With this node you can apply dimension lines, guide lines and the dimension values to a document containing artwork for a bag.

There are different kinds of bags: gusseted bags, paper bags, pillows and sachets. Each type of bag exists in different sizes, but the initial structure remains, independent of type and/or size. Before a bag is processed into its final shape, it is a rectangular shape on which the artwork is positioned. To know where certain artwork needs to be positioned a technical representation can be useful. This technical representation will give you an insight of the location of the different parts of the bag. Therefore, dimension lines can be placed on the artwork.

With Dimensions, you can apply dimension lines, guide lines and the dimension values to a document containing artwork for a bag. These dimensions are placed according to a reference box and are based on a certain type of bag. The bag divisions will automatically take the sizes of the reference box into account when they are being defined.

The node has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

## **Global Settings**

- Reference: here you can define the reference box and the position of the dimension lines within the reference box.
- **Dimension above / below / to left of / to right of** buttons: with these buttons you can define where the dimension lines of the reference box should be placed.
- Offset input field: here you can define a value that determines the distance between the border of the reference box and the dimension line.
- Generate border on box button: with this button you can create a dashed border on the reference box.
- Stroke Width: here you can define the thickness of the dashed lines and the width of the lines at the end of the arrows in case arrows are defined.

## Bag

- **Horizontal/Vertical** buttons: if you select the icon(s), the parameters for the horizontal/vertical orientation become available. When you activate the horizontal button, the width of the reference box is used, when you activate case the vertical button, the height of the reference box is used.
- **Type**: here you can select a predefined parameterized type of bag. You can extend this list by adding your own custom bags. In that case you need to enter a definition in the **Custom Bag Definition** field. User defined bags are indicated by a U in front of the type name.

# Example of a custom bag definition

B;R;S;2\*R;S;R;E

Where:

- **B**: (begin) refers to the left or top of a bag, depending on the orientation.
- E: (end) refers to the right or bottom of a bag, depending on the orientation.
- S: refers to the side of a bag.
- **R**: corresponds with the result parameter. Only this parameter can be used in combination with a multiplication factor, for example **2**\***R**.
- **Begin** / **End** / **Side** input fields: here you can define the begin, end and side dimensions in case of a horizontal orientation. When the orientation is vertical, these parameters correspond with the top, bottom and side of the bag. Depending on the type of bag, the Side input field might not be available.

# **Horizontal Division**

Based on the type of bag selected from the **Type** drop-down and the length (the width or height of the reference box), the divisions for the horizontal orientation are calculated from the parameters in the **Bag** section and displayed here.

- **Position**: with the segmented buttons **Dimensions above** box / **Dimensions below** box you can define the position of the dimension lines and their values.
- Offset: here you can define the distance between the reference box and the placed dimension.

# **Vertical Division**

Based on the type of bag selected from the **Type** drop-down and the length (the width or height of the reference box), the divisions for the vertical orientation are calculated from the parameters in the **Bag** section and displayed here.

- **Position**: with the segmented buttons **Dimensions to left of** box / **Dimensions to right of** box allows you can define the position of the dimension lines and their values. The distance between the reference box and the placed dimension are specified in the Offset fields.
- Offset: here you can define the distance between the reference box and the placed dimension.

## **Dimension Arrows**

With Dimension Arrows you can control how the arrows that are created when creating dimension lines.

A Dimension Arrow is a group containing:

- A line with markers on both ends.
- A dimension value in a particular unit, centered on the line.
- A unit (optional).
- Font: here you can define the font family.
- Font Style: here you can define the font style.
- Font Size: here you can define the size.
- Unit: here you can define the unit. Options: Millimeter (mm), Centimeter (cm), Inch (in) and Point (pt).
- Unit Precision: here you can define the unit precision.
- Include Unit Symbol: if you select this checkbox, the unit symbol will be included.
- Marker: here you can select the type of marker from a drop-down list.
- Stroke Width: here you can define the stroke width.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### **Export as CIP**

With this node you can export a *Print Production Format File (PPF)* containing the ink-key presets for printing presses.

It has the following configuration options:

- Input File: here you can specify the input file. Select the pencil to open Expression Builder.
- Output Name: here you can specify the output name.
- Resolution: here you can define the resolution for the generation of the image preview within the PPF file.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### Export as Image

With this node you can create an RGB JPEG image of the current page of the open PDF document.

It has the following configuration options:

- Input File: here you can select the input file. Select the pencil to open Expression Builder.
- Output Name: here you can select an output name for your file.
- **Resolution**: here you can define the resolution for the image.
- **Compression**: here you can define the JPEG compression. You can enter a value from 1 to 10. When you enter **1**, the output is of low quality (where compression artifacts are very visible). When you enter **10**, the output is of high quality (where compression artifacts are less visible).
- from the drop-down list in order to reduce the file size. The options are JPEG Low Quality, JPEG Medium Quality and JPEG High Quality. When you select Low, the compression rate is high which will result in low quality images but small file sizes. When you select High, the compression rate is low with less quality loss but the file size is bigger.
- List of Separations: here you can specify which separations should be output by entering a separation name. You can enter several separation names using a comma separated list.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### **Export Die**

With this node the trim paths of any kind of Step and Repeat created in PACKZ will be exported to a PDF file.

It has the following configuration options:

- **Input File**: here you can specify the input file with the trim paths.
- **Output File**: here you can specify the output PDF file where the trim paths will be exported to. The exported trim paths are 100% Black strokes, with no fill color.

### **Export Gravure**

With this node you can output PDF step and repeat files as 8-bit TIFF files for Gravure Printing.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Mode: here you can define the output mode. Options:
  - Separated: if you select this option, the output will be separated TIFF files.
  - Composite: if you select this option, the output will be a composite TIFF file.
  - **OHIO separated**: if you select this option, the output will be a multipage PostScript file (one page per separation) to be used on OHIO Collage systems. The file will be flattened first and then does a limited PostScript export, exporting only flat paints and images. Everything is vectorized and all other paints are converted to images.
- **Oversampling**: here you can select the *oversampling* ratio. For example, if the resolution is set to 300 ppi and the oversampling ratio to 2, the images will be converted at 600 ppi and downsampled to 300 ppi.
  - **Oversampling All**: if you select this checkbox, the complete document will be oversampled, including marks. If you leave this checkbox unselected, only the step and repeat stations will be oversampled.
- Set Page Curve: here you can enter the name of a *curve* that will be applied to the page during RIP phase.
  - Apply Curve: when you select this checkbox, the curve defined in the field Set Page Curve will be applied to the artwork objects on the design to make it visible.

**Note:** If you cmd+alt+click the checkbox, you can use Expression Builder to specify the name of a screening from the saved screening list in the shared folder.

- **Resolution**: here you can define the resolution of the TIFF files.
- Noise for Shadings: here you can define how much noise should be applied in the shadings to prevent them from banding. The higher the value, the more noise is added.
- **Pagebox**: here you can define the page box that will define the dimensions of the TIFF files.
- Rotation: here you can select the rotation of the TIFF files. The files can be rotated 180°, 90° CW or 90° CCW.
- Compression: here you can select the compression of the output file.
- **Fast Forward Data**: if you select this checkbox, an extra SVS file with the same name as the tiff name will be output. This option is available for both composite and separated output.

### Separations

- All: if you select this checkbox, all the separations of the PDF will be output as a separate TIFF file. If you select/ deselect the checkboxes per separation, you can select which separations should (not) be output as TIFF.
- **Process and Spot Separations**: if you select/deselect this checkbox, the process and spot separations will be selected/deselected.
- Technical Separations: if you select/deselect this checkbox, the technical separations will be selected/deselected.
- Varnish Separations: if you select/deselect this checkbox, the varnish separations will be selected/deselected.

- **Include separations**: here you can enter the separations that will be exported. You can use a comma separated list. If you leave this field empty, all separations will be exported.
- Mirror all: when you select this checkbox, the TIFF file(s) will be mirrored vertically.

**Note:** Mirror is applied after rotation when selected.

- Mirror Separations: here you can enter the separations that need to be mirrored.
- Distort Separations: here you can enter the separations that need to be distorted.
- File Naming: here you can select the naming method. Options:
  - Separation Name: if you select this option, the output TIFF files will contain the PDF filename followed by an underscore (\_) and the separation name.
  - Separation Index: if you select this option, the output TIFF files will contain the PDF filename followed by a hash tag (#) and the index number of the separation.

### Distortion

- Horizontal Distortion: here you can define a horizontal distortion. Options:
  - None: if you select this option, no distortion will be applied.
  - **Percentage**: if you select this option, the horizontal distortion is taken from a percentage.
  - Horizontal Distortion: Percentage: here you can enter the percentage of the horizontal distortion.
  - From Page: if you select this option, the distortion value as set in the Assets: Output palette will be taken over.
  - K Factor: of you select this option, the horizontal distortion is taken from the K Factor.
  - Horizontal Distortion: K Factor: here you can enter a K-Factor (thicK-Factor) which defines the thickness of the plate and the mounting tape. This can be a positive or a negative number. A positive number indicates horizontal dimension increase, a negative number indicates a horizontal dimension decrease.
- Vertical Distortion: here you can define a vertical distortion. Options:
  - None: if you select this option, no distortion will be applied.
  - **Percentage**: if you select this option, the vertical distortion is taken from a percentage.
  - Horizontal Distortion: Percentage: here you can enter the percentage of the vertical distortion.
  - From Page: if you select this option, the distortion value as set in the Assets: Output palette will be taken over.
  - K Factor: of you select this option, the vertical distortion is taken from the K Factor.
  - Vertical Distortion: K Factor: here you can enter a K-Factor (thicK-Factor) which defines the thickness of the plate and the mounting tape. This can be a positive or a negative number. A positive number indicates vertical dimension increase, a negative number indicates a vertical dimension decrease.

### Example

If the job has a vertical dimension of 120.000 mm and the vertical distortion based on the K Factor is -1.000 mm, the resulting tiff will have a vertical dimension of 119.000 mm.

**Note:** Distortion is always applied after rotation.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

# Export PACKZ Metadata

With this node you can export the selected PACKZ variable to a CLOUDFLOW variable.

It has the following configuration options:

• Input File: here you can define the input file.

- PACKZ Variable: here you can define the PACKZ variable you want to convert.
- **Cloudflow Output Variable**: here you can define the CLOUDFLOW variable to which you want to convert the PACKZ variable.
- **Template File**: here you can define the template that will be used to export the metadata of the file. See the PACKZ Reference manual for more information on how to create templates.
- **Output File**: here you can define the output path to which the assets file should be exported.
- **Note:** See the PACKZ Reference manual for more information on PACKZ variables.

### Flatten

With this node you can flatten a file.

Flattening is the process of removing *blend modes*, opacities and overprints by replacing them with a normal object so that the PDF document preview remains the same, but applications not supporting transparency can handle these files.

The node has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

### General

• Flatten Overprints: if you select this checkbox, overprints are flattened.

### Images

• Maximum Resolution: here you can define a maximum resolution when flattening is needed between images, shadings and/or line work, avoiding heavy files.

## Shadings

- **Resolution**: shadings are converted to images during the flattening process. In this field you can define a resolution for the images.
- Noise: here you can define how much noise should be applied to the shadings when converting to image to prevent them from banding on the image setter or proofer.

However, noise may cause screen defects when using hybrid screens in which conventional and frequencymodulated screens are combined. Therefore noise can be applied in certain ranges, between certain percentages:

- Apply from 0% to ...%
- And from ...% to 100%
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### **Grid Imposition**

With this node you can create an imposition of different PDF files with the same dimensions and gathered in the same folder. A layout file can define the imposition as well.

It has the following configuration options:

- Input File(s): here you can define the input file(s). Select the pencil to open Expression Builder.
- Output File: here you can define the output file. Select the pencil to open Expression Builder.
- Layout File (optional): here you can define the layout file. A Grid layout file specifies how the stations are laid out in the grid.

# Plate

- Width: here you can define the plate width.
- Height: here you can define the plate height.

## Paper

- Width: here you can define the paper width.
- Height: here you can define the paper height.
- **Reference Point**: here you can define the position of the paper on the plate, taking the **horizontal** and/or **vertical** paper offset values into account.
- Horizontal Offset: here you can define the horizontal paper offset.
- Vertical Offset: here you can define the vertical paper offset.

## Imposition

- **Reference Point**: here you can define the position of the complete repetition on the paper, taking the **horizontal** and/or **vertical** offset values into account.
- Horizontal Offset: here you can define the horizontal imposition offset.
- Vertical Offset: here you can define the vertical imposition offset.
- Make Placeholder: if you select this checkbox, a placeholder is used.

## Padding

- Left padding for this block: here you can define a clear area on the left between the repetition and the Trim Box.
- **Right padding for this block**: here you can define a clear area on the right between the repetition and the Trim Box.
- Top padding for this block: here you can define a clear area on top between the repetition and the Trim Box.
- **Bottom padding for this block**: here you can define a clear area at the bottom between the repetition and the Trim Box.

## Repetition

- **Page Number**: in case a multipage PDF is loaded as a station, by default the first page is selected. To change this, you can define a different page by entering a page number.
- Trim Path: here you can define the area of the station that will be used to create the repetition.
- Orientation: here you can define the orientation of the station (up, down, left, right).
- Horizontal Count: here you can specify the number of horizontal stations.
- Vertical Count: here you can specify the number of vertical stations.
- **Horizontal and Vertical Step**: here you can define the distance from one station to the other within the step and repeat (that is, the distance from the top left corner from one station to the top left corner of the next station).
- Gap: here you can specify the horizontal white gap between the stations of a step and repeat block.
- Vertical Gap: here you can specify the vertical white gap between the stations of a step and repeat block.
- **Origin**: here you can define the starting point of the imposition.
- **Disposition**: here you can define the direction how the stations will be positioned.

## Station numbers

- Generate Station Numbers: if you select this checkbox, station numbers will be generated, which means that a unique number is placed on every station of a step and repeat block. This allows to easily recognize an error on a specific station of a printed step and repeat.
- **Prefix**: here you can define a prefix to the station numbers.
- Style: here you can define whether the station numbers should be written as numbers or letters.

## Bleed

• Bleed: here you can define the amount of *bleed* that should be added to the step and repeat.

- Get bleed distance from station: if you select this checkbox, the bleed distance as defined in the station will be taken which will overwrite the value set in the **Bleed** input field.
- **OneUp File List Parameters**: here you can define a variable corresponding to each 1up defined in the above file list. The variable consists of a JSON list of parameter objects. Each item corresponds to an item at the same position in the above **OneUp File List**.

## Handle Separations

With this node you can remove, remap, change the order or update the colorimetric information of separations in a PDF document.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- **Apply changes on selection only**: if you select this checkbox, the changes will only be applied to a predefined selection (for example, a selection specified with the **Layers** node).

# Add Separation

- Name: here you can define the separation name to be added. You can enter several separations with a comma separated list (for example Die, Pantone 123 C, K). The separation names are case sensitive.
- Type: here you can define a separation type to add all separations of certain type.
- Color Type: here you can define a color type.
- Color Value: here you can define a color value.

### **Remove Separation**

- Name: here you can define the separation name to be removed. You can enter several separations with a comma separated list (for example Die, Pantone 123 C, K). You can also use a wild card \* (for example when you enter **PD**\*, all separation names starting with PD will be removed). The separation names are case sensitive.
- Type: here you can define a separation type to remove all separations of certain type.

## Example

If you want to remove all technical separations, select **Technical** from the **Type** drop-down list and leave the **Name** file empty. If you want to remove the both technical and varnish separations, you need to do that in two **Handle Separation** nodes. The node is memory-optimized so using two nodes will not make and performance problems.

• All Unused: if you select this checkbox, all unused separations will be removed from the document.

## **Remap Separation**

- Name: here you can enter a separation name that you want to remap. You can use a wild card, however, you cannot use a comma separated list.
- **Type**: here you can specify a type of separation that you want to remap. If the type is other than **Any**, only separation of the specified type will be remapped.



**Tip:** If you want to remap all separations of a specific type, you can put an asterisk to in the **Name** field (\*).

- to CMYK: if you select this checkbox, the field To Name is ignored. All matching separations will be mapped to CMYK.
  - *i* **Tip:** If you want to map all non-CMYK separations to CMYK, you can enter \* in the **Name** field. The asterisk will match all separations, including C, M, Y, and K, but these will not be mapped to CMYK since they already are CMYK.
- **To Name**: here you can define the separation name to be mapped to. If the specified name is not present in the input PDF, the separation will be taken from the active Color Books.
- To Type: here you can specify the type where all matching separations will be mapped to.

• **Rename if does not exist**: if you select this checkbox, the source separation that you want to remap will be renamed in case the target separation does not exist.

## Example

If you want to remap all separations containing string 817 to Technical:

- 1. Set Remap Separations, Name: to 817
- 2. Set Remap Separations, Type to Any
- 3. Set Remap Separations, To Name to empty
- 4. Set Remap Separations, To Type to Technical
- Mark as darkest: here you can set the selected separation as darkest color. In other cases, when there is Black present in the file, Black will always be marked as automatic darkest color.

## **Separations Order**

This section allows you to perform certain actions concerning the separation order.

- Order: here you can specify an order for the separations in a JSON formatted way (for example ["Cyan", "Magenta", "Yellow", "Black"]. The default value already specifies the ink order for CMYK separations.
- Action: here you can specific the action to be taken. Options:
  - No Action: if you select this option, no action will be taken, regardless of the existing separation order.
  - Error If Order Different: if you select this option, the node will generate an error when the separation order in the document is different from the separation order specified in the field Order.
  - Set The Order: if you select this option, the separation order will be changed to the order specified in the field Order.
- Reverse Order of All Inks: if you select this checkbox, the separation order in the document will be reversed.

## **Color Book**

- Ink Book Name or Cloudflow URL: here you can specify the ink book name or URL using a CLOUDFLOW path.
- Action: here you can specify the action to be taken. Options:
  - No action: if you select this option, no action will be taken.
  - Update All Inks In Order: if you select this option, all inks mentioned in Separations Order will be updated according to the selected ink book.
  - Update All Inks from Remap Section: if you select this option, all inks mentioned in Remap section will be updated according to the selected ink book.
  - Update All Inks In Document: if you select this option, all inks in the document will be updated according to the selected ink book.

### **Printing Methods**

With Printing Methods you can define a printing method to a separation.

- Name: here you can define the name of the separation to define the printing method to.
- Type: here you can define the printing method.

Select + to add a printing method.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### **Insert Path**

With this node you can insert a path in a document.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Page Range: here you can select the page range for the path. Options:
  - All: if you select this option, the path will be added for all pages.
  - First: if you select this option, the path will be added for the first page only.
  - Selected Range: if you select this option, the path will be added for specific pages only.
- Separation Name: here you can define the separation of the path.
- Ink Value: here you can define the ink value of the path. For example if you enter 100, the path will be created in a 100% ink value.
- Stroke Width: here you can define the stroke width of the path.
- Layer Name: here you can define on which layer the path should be created.
- **Path Points**: here you can define the path coordinates.
  - In case of single path, the input needs to be a single flat list containing points: [ x, y, c1x, c1y, c2x, c2y, x, y, c1x, c1y, c2x, c2y, ...]
  - In case of compound path, in needs to contain a list of lists, each sub-list represents a sub-path of compound path: [ [ x, y, c1x, c1y, c2x, c2y...], [ x, y, c1x, c1y, c2x, c2y...]...]

Where:

- x<sub>n</sub>, y<sub>n</sub> corresponds to the starting point of n-th segment.
- c1x<sub>n</sub>, c1y<sub>n</sub> corresponds to the bezier control point of beginning of n-th segment, relative to x<sub>n</sub>.
- c2x<sub>n</sub>, c2y<sub>n</sub>: corresponds to the bezier control point of ending of n-th segment, relative to y<sub>n</sub>.

And:

- Every coordinate is relative to x<sub>n</sub>, y<sub>n</sub>.
- Every next segment coordinate is relative to the previous x(n-1), y(n-1).
- **Note:** In many cases, the coordinates will be defined in a variable through prior scripting nodes.

### Example

If you want to add a path with 50% paint and 0.1 mm width to a PDF file, you need to add the **Script** node prior to the Insert Path node. If you define the following script in the Script node, the points will be converted into a string and communicated as a string of values to the Insert Path node:

```
setResults ( {
variables : {
  abs: [150, 230, 0, 0, 0, 0, 0, 280, 0, 5.518, -5.518, 0]
}
}
```

- **Reference**: here you can define to which reference point the path is relative. Options:
  - Absolute: if you select this option, the position of the path is relative to the document's absolute zero point.
  - Media Box: if you select this option, the path is relative to the left-bottom point of the document's media box.
  - Crop Box: if you select this option, the path is relative to the left-bottom point of the document's crop box.
  - Bleed Box: if you select this option, the path is relative to the left-bottom point of the document's *bleed box*.
  - Trim Box: if you select this option, the path is relative to the left-bottom point of the document's *trim box*.
  - Art Box: if you select this option, the path is relative to the left-bottom point of the document's *art box*.
  - Selection Bounds: if you select this option, the path is relative to the bounding box of a selection.
  - Extra Horizontal Offset: here you can define extra horizontal offset to create an extra shift in the path.
- Extra Vertical Offset: here you can define extra vertical offset to create an extra shift in the path.

- **Preserve CAD Groups in CF2**: if you select this checkbox, no groups will be created for sub routines in the CFF2 when importing repetition CFF2 files.
- **Remove Text in CF2**: if you select this checkbox, all the text will be removed, except text that has been marked as a dimension.
- Remove dimensions in CF2: if you select this checkbox, all the text and lines that are marked as dimension will be removed.
- Plate/Drum from CFF2: here you can select the Line Types that set the Plate/Drum. Options
  - None: if you select this option, predefined Line Types will be used to set the Plate/Drum, not line types from CFF2.
  - LL-UR: if you select this option, the lower left (LL) and upper right (UR) defined in the CFF2 file sets the Plate/Drum of the repetition.

When a line type is selected, CLOUDFLOW will calculate the bounding box of all the straight lines (excluding subroutines) and use the dimensions of that box as the Plate/Drum Size.

- Printable Area from CFF2: here you can select the Line Types that set the Printable Area. Options:
  - None: if you select this option, predefined Line Types will be used to set the Printable Area, not line types from CFF2.
  - LL-UR: if you select this option, the lower left (LL) and upper right (UR) defined in the CFF2 file sets the Printable Area of the repetition.

When a line type is selected, CLOUDFLOW will calculate the bounding box of all the straight lines (excluding subroutines) and use the dimensions of that box as the Printable Area.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

## Layers

With this node you can add or remove layers from the PDF document or select objects on layers to perform actions on them.

It has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

# Operations

- **Remove Empty Layers**: if you select this checkbox, all empty layers will be removed.
- **Remove Layer Names**: here you can define the layer names of the layers that you want to remove. If you want to remove several layers, you can separate the layer names with a comma. The layer names can be specified with a wild card. For example, if you put **Layer**\*, all layers starting with the word **Layer** will be removed.
- Add New Layer on Top: here you can define the name of a layer you want to add to the existing layers in the document. The layer will be placed on top.
- Set Layers Visible: here you can define the layer names of the layers that you want to set visible.
- Set Layers Hidden: here you can define the layer names of the layers that you want to set hidden.
- Set Layers Printable: here you can define the layer names of the layers that you want to set printable. These layers will be output.
- Set Layers Not Printable: here you can define the layer names of the layers that you want to set non printable. These layers will not be output.
- Set Layers Locked: here you can define the layer names of the layers that you want to set locked.
- Set Layers Unlocked: here you can define the layer names of the layers that you want to set unlocked.
- Strip Acrobat Layers: here you can select the Acrobat layers that you want to strip. The content of these layers will be not be removed but converted into a regular group.

# Selection

- Clear Selection: if you select this checkbox, all selections (for example from a previous node) will be cleared.
- Select Layer names: here you can specify one or more layers to select. You can separate several layer names with a comma. The layer names can be specified with a wild card. For example, if you put Layer\*, all layers starting with the word Layer will be selected.
- **Deselect Layer names**: here you can specify one or more layers to deselect. You can separate several layer names with a comma. The layer names can be specified with a wild card. For example if you put **Layer**\*, all layers containing with the word **Layer** will be deselected.
- Set Active Layer Name: here you can define the layer to activate. The active layer is the layer on which new objects will be added.
- Move Selection to Layer Name: here you can specify the layer to where the selection should be moved.

# **Rename Layers**

- From Names: here you can select one or more layers that should be renamed. You can separate several layer names with a comma.
- To Names: here you can define the new name(s) of the layer(s) to be renamed. When renaming several layers, make sure to use the same order as the layers you specified in the field **From Names**.

# Layers Order

- **Put the Layers on Top**: here you can define one or more layers to place on top. You can separate several layers with a comma. In that case, the first layer of the list will be placed on top, followed by the second one in the list etc.
- **Put the Layers on Bottom**: here you can define one or more layers to place on the bottom. You can separate several layers with a comma. In that case, the first layer of the list will be placed on the bottom, followed by the second one in the list etc.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

# **Proc Steps**

- Layer name: here you can select the layer to assign the Proc Steps to.
- **Groups**: here you can select the Groups.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

## Live Objects

With this node you can create active information panels, dynamic registration marks and live designs that automatically update depending on PDF document content or data from external resources.

It has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

- File: here you can define the Live Object PDF template created in PACKZ.
  - **Note:** You need **PACKZ** to create Live Object PDF templates.
- JobID: here you can select a JobID.
- Page Range: here you can select the page range where the live objects should be created. Options:
  - First: if you select this option, the live objects will be added to the first page only.
  - All: if you select this option, the live objects will be added to all pages.
  - Single: if you select this option, the live objects will be added to a specific page only.
  - On New Page: if you select this option, a new page will be added where the live objects will be added to.
- Layer Behavior: here you can define how layers should behave when importing marks files. Options:
  - Insert in Marks Layer: if you select this option, the marks will be put in a Marks layer.
  - Preserve Layers: if you select this option, the layer structure of the marks file will be preserved.
  - **Merge with Existing Layers**: if you select this option, the layer structure of the marks file will be merged with existing unlocked layers with the same name in the document whenever possible. In case of multiple layers with the same name, the top or bottom layer will be used depending on the selected option in the parameter **Position**.
- **Position**: here you can define the position of the live objects in the layer. Options:
  - Top: if you select this option, the layer will be added to the top of the Object Tree.
  - Bottom: if you select this option, the layer will be added to the bottom of the Object Tree.
- Update All Live Objects: if you select this checkbox, all live objects will be updated according to the selected input file.

# Positioning

- Source Page Number: here you can define the source page number in case the placed file is a multipage PDF.
- **Positioning Reference**: here you can define the positioning reference for the inserted PDF file. Options:
  - **Do not position**: if you select this option, the PDF will be positioned using the PDF absolute coordinates as a reference.
  - **Ruler**: if you select this option, the PDF will be positioned using ruler position that is set in PACKZ as a reference.
  - Media Box: if you select this option, the PDF will be positioned using the media box as a reference.
  - Crop Box: if you select this option, the PDF will be positioned using the crop box as a reference.
  - Bleed Box: if you select this option, the PDF will be positioned using the bleed box as a reference.
  - Trim Box: if you select this option, the PDF will be positioned using the trim box as a reference.
  - Art Box: if you select this option, the PDF will be positioned using the art box as a reference.
- **Point**: here you can define which point of the box is the reference point. For example, when the **Positioning Reference** is set to **Media Box**, the **Point** can be set to the center, the lower left, the upper right... of the Media Box.
- **Positioning Target**: here you can define the positioning target of the PDF file. This is similar to the parameter **Positioning Reference**, but in this case, the position is defined in the input PDF document.
- **Point**: here you can define which point of the box is the positioning point. The PDF is placed in such a way that the **Reference Point** and the **Target Point** are on the same position, except if you define an **Extra Offset Horizontal** or an **Extra Offset Vertical**.
- Extra Offset Horizontal: here you can specify an extra horizontal offset between the Reference Point and the Target Point.
- Extra Offset Vertical: here you can specify an extra vertical offset between the Reference Point and the Target Point.

## Preferences

- Units: here you can define the unit preferences.
  - Length: here you can specify the length units. If you select Default, the units in the CLOUDFLOW User Preferences are used.
  - Area: here you can specify the area units. If you select **Default**, the units in the CLOUDFLOW User Preferences are used.
- Languages: here you can define the language preferences.
  - Language: here you can define the language that will be used for the PDF template that is used to generate the report in PACKZ containing the Analyze and Fix results.
- Ink Consumption: here you can define which area is taken into account for calculating the ink consumption that Live Objects will show.
- Use trim path when available : if you select this checkbox, the trim path, if present, is taken into account when calculating the ink consumption.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

# Nested Step And Repeat

With this node you can step and repeat packages based on a CAD file (CFF2 file) that contains repetition information.

It has the following configuration options:

- Input file: here you can define the input file. Select the pencil to open Expression Builder.
- Create an empty placeholder: if you select this checkbox, an empty placeholder will be created, that can later be replaced.

## Plate

- Width: here you can define a custom plate width.
- Height: here you can define a custom plate height.

# Paper

- Width: here you can define a custom paper width.
- **Height**: here you can define a custom paper height.
- **Reference Point**: with the reference point you can define the position of the paper on the plate, taking the horizontal and/or vertical paper offset values into account. The horizontal and vertical paper offset values can be defined in the fields **Horizontal Offset** and **Vertical Offset**.
- Horizontal Offset: here you can define the horizontal offset that is taken into account when selecting a reference point for placing the paper on the plate.
- Vertical Offset: here you can define the vert ical offset that is taken into account when selecting a reference point for placing the paper on the plate.

# Step And Repeat

- **Reference Point**: with the reference point you can define the position of the complete repetition on the paper, taking the horizontal and/or vertical offset values into account. The horizontal and vertical paper offset values can be defined in the fields **Horizontal Offset** and **Vertical Offset**.
- Horizontal Offset: here you can define the horizontal offset when defining a reference point for the repetition on the paper.

• Vertical Offset: here you can define the vertical offset when defining a reference point for the repetition on the paper.

## Select CAD file

- CAD File: here you can select the CAD file that will be used to create the nested step and repeat.
- Transform: here you can select the transform type. Options:
  - None: if you select this option, no transformation will take place.
  - **Create Verso**: if you select this option, a new document (**filename\_verso.pdf**) will be created in which the repetition is horizontally mirrored. This is also called **Turn**. The stations of the step and repeat are all placeholders.
  - **Create Tumble**: if you select this option, a new document (**filename\_tumble.pdf**) will be created in which the repetition is vertically mirrored. The stations of the step and repeat are all placeholders.
  - Create Rotated 180°: if you select this option, a new document (filename\_Rotated 180.pdf) will be created in which the repetition is rotated 180°. The stations of the step and repeat are all placeholders.
- Transparent Stations: here you can define the station transparency. Options:
  - From Station: if you select this option, the transparency is taken from the station.
  - **Transparent**: if you select this option, all the stations in the repetition are placed in darken mode, allowing bleed in overstepping blocks.
  - Knockout: if you select this option, a knockout object will be created underneath all the stations.
- **Import CFF2**: if you select this checkbox, the CAD information is imported with the specified separation and stroke width. By default the first technical separation or black is selected. You can also specify a separation name. The system will look for the entered separation name in all color books and will use (and add) the color book separation. If there are more than two matches, then the first one found will be used. When entering a non-existent separation name in the drop- down field the separation will be created. If no stroke width is defined, the width from the CFF2 file is taken. The interpreted CAD information is converted to open paths with a stroke width, in the defined separation in overprint mode. They are imported in a new layer. This layer will have the name specified in the CFF2 file and will contain sub layers: Cut Lines, Crease Lines and other sub layers depending on the structure of the CF2-file.

If you do not select this checkbox, the CFF2 file will be analyzed to create a trim path from the die information. Every subroutine in the CFF2 file will be analyzed only once. This will ensure that all the stations using the same subroutine are identical.

- Use CFF2 Placeholder names: here you can select the placeholder names. Options:
  - **Default**: if you select this option, the placeholders will get a default name **Placeholder XX**, where XX is a consecutive number, unless you have specified a different name in the file **config.txt**. Only use the **Default** option for backwards compatibility.
  - Use CLOUDFLOW subroutines: if you select this option, the names of subroutines will be used as a placeholder name in the loaded Step and Repeat CF2 file.
  - Use Placeholder with order number: if you select this option, the placeholders will get a default name Placeholder XX, where XX is a consecutive number.
- Plate/Drum from CFF2: here you can select the Line Types that set the Plate/Drum. Options
  - None: if you select this option, predefined Line Types will be used to set the Plate/Drum, not line types from CFF2.
  - LL-UR: if you select this option, the lower left (LL) and upper right (UR) defined in the CFF2 file sets the Plate/Drum of the repetition.

When a line type is selected, CLOUDFLOW will calculate the bounding box of all the straight lines (excluding subroutines) and use the dimensions of that box as the Plate/Drum Size.

- Printable Area from CFF2: here you can select the Line Types that set the Printable Area. Options:
  - None: if you select this option, predefined Line Types will be used to set the Printable Area, not line types from CFF2.
  - LL-UR: if you select this option, the lower left (LL) and upper right (UR) defined in the CFF2 file sets the Printable Area of the repetition.

When a line type is selected, CLOUDFLOW will calculate the bounding box of all the straight lines (excluding subroutines) and use the dimensions of that box as the Printable Area.

## Repetition

- **Placeholder Name**: here you can select the placeholder in the step and repeat palette that will be replaced by a PDF document. See the **PACKZ** Reference Manual for more information on placeholders in a nested step and repeat palette.
- **Page Number**: in case a multipage PDF is loaded as a station, by default the first page is selected. To change this, you can define a different page by entering a page number.
- Reference: here you can define the reference area of the station that will be used to create the step and repeat.
- Orientation: here you can define the orientation of the station (up, down, left, right).
- Mirror: here you can define whether the station needs to be mirrored (horizontally or vertically).

### Station numbers

- Generate Station Numbers: if you select this checkbox, station numbers will be generated, which means that a unique number is placed on every station of a step and repeat block. This allows to easily recognize an error on a specific station of a printed step and repeat.
- **Prefix**: here you can define a prefix to the station numbers.
- Style: here you can define whether the station numbers should be written as numbers or letters.
- Numbering Origin: here you can define where the numbering should start.
- Flow: here you can define in which direction the numbering should go.
- Bleed: here you can define the amount of *bleed* that should be added to the step and repeat.

#### Bleed

- Get flap information from station: if you select this option, the Flaps setup as defined in the station PDF is taken.
- Bleed: here you can define the bleed distance in the station file.
- Get bleed distance from station: if you select this option, the bleed distance is taken from the bleed distance in the station file that is defined in PACKZ.
- Left flap width, Right flap width, Top flap height, Bottom flap height input fields: here you can specify the length of the flaps at the left / right / top and bottom of unrotated stations.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

**Note:** You can check the result of the nested step and repeat and change it if needed in **PACKZ**.

### Create repetition from a JSON layout file

You can create a repetition from a JSON layout file. To do this, configure the Nested Step&Repeat node in the following way:

- 1. Set all the parameters to the default value, except for these ones:
  - **Input File**: here you need to specify the PDF reference file which will be used for the output reference. You can point to the input JSON file.
  - Select CAD file > CAD File: here you need to specify the input JSON layout file.
  - **Output**: you need to customize this parameter, if not, your output PDF will have the extension .json.

#### Packzimizer

With Packzimizer you can merge multiple artwork files onto a predefined sheet in the most economic way (based on the number of requested copies).

The files are positioned in the most efficient way in order to waste as little sheet material as possible. The way to position the files is based on the cut path. If the file does not have a cut path, the trim box will be used.

The output is an imposition for a digital printing press and a die cut form for a laser cutter.

It has the following configuration options:

• Input File test: here you can define the input file(s).

#### Sheet

- Width: here you can define the width of the sheet.
- Height: here you can define the height of the sheet.
- Allowed Rotation: here you can define the allowed rotation of the file(s).
- Mode: here you can define the mode. Options:
  - **for Digital**: if you select this option, the best solution will be searched for so that the required quantity is reached for every added station resulting in a Nested Step and Repeat on every generated page, taking the parameters set into account.
  - **for Traditional**: if you select this option, a PDF file will be created with one page containing an optimal combination of all added stations in a Nested Step and Repeat, where a mutual ratio between the entered wanted numbers is respected as much as possible.
- Allow different shapes on one sheet: if you select this checkbox, different files shapes can be positioned on one sheet. If you leave this checkbox unselected, only one file shape is positioned on the sheet.
- Start Position: here you can define the start position of the files.
- Gap: here you can define the gap between the file shapes.
- Center Layout: if you select this checkbox, the layout with the files will be centered on each sheet.

### Repetition

- **Page Number**: here you can define the page number of the input file that needs to be positioned on the sheet.
- Count: here you can define the number of files you need.
  - **Note:** If you want to create an imposition that exists of different numbers of different files, you'll need to build a workflow with a script that collects all the files from a certain folder. This is an example of a possible (template) script:

```
// var files[] is the list of all the file urls
// var page[] is the list of pagenumbers for each file
// var count[] is the list of counts for each file
// var orientation[] is the allowed rotation for each file
var ui_params = [];
for (int i = 0; i < files.length; ++i)
{
    var ui_param = {
        "file": files[i].replace(/.*\//,""),
        "page": page[i],
    }
}</pre>
```

```
"count": count[i],
    "orientation": orientation[i]
  };
  ui_params.push(ui_param);
}
var params = {
    "page":page.join(),
    "count":count.join(),
    "orientation":orientation.join()
};
setResultVariables({
    "params":params,
    "ui_params":ui_params
});
```

Output: here you can define the output path. Select the pencil to open Expression Builder.

## Pageboxes

With this node you can define the exact size of a PDF document.

It has the following configuration options:

- Input File: here you can select the input file. Select the pencil to open Expression Builder.
- **Reference Point**: here you can define an anchor reference point that defines in which direction the page box will be modified. For example a top left point indicates that the top left corner of the page box will be unmodified and that all changes will be made towards the bottom and the right side.
- Change pagebox orientation: here you can change the orientation of the page box from portrait to landscape or visa versa.

### Set Pagebox(es)

- Set: here you can select the page box of which you want to adjust the dimensions. For more information on page boxes, see Page boxes.
- To: here you can select the page box to which you want to adjust the dimensions.

For example Set Trim Box to Media Box will adjust the dimensions of the Trim Box to the Media Box.

### Add to Pagebox(es)

- To: here you can select the page box of which you want to change the dimensions.
- Width: here you can enter a value to define how much the width of the selected page box should expand.
- Height: here you can enter a value to define how much the height of the selected page box should expand.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

## Prepress

With this node you can adjust the content of a PDF document based on a page box.

It has the following configuration options:

- Input File: here you can specify the input file. Select the pencil to open Expression Builder.
- Apply Changes On Selection Only: if you select this checkbox, the changes will only be applied to a predefined selection (for example, a selection specified with the Layers node).

- Clip on Page Box: here you can select a page box where the content of the PDF will be clipped on. This means that all information outside the selected page box will be clipped away.
- **Execute Clipping**: if you select this checkbox, all hidden and unnecessary information resulting after the creation of a clipping path is removed. This means that all paths outside the clipping mask will be clipped away.
- Set Page Curve: here you can enter the name of a *curve* that will be applied to the page during RIP phase.
- Set Page Curve from File: here you can browse to a JSON formatted curve file.
- Apply Curve: if you select this checkbox, the curve defined in the Set Page Curve field will be applied to the artwork objects on the design to make it visible.

**Note:** If you cmd+alt+click the checkbox, you can use Expression Builder to specify the name of a screening from the saved screening list in the shared folder.

- Apply Screening: here you can select a *screening* that should be applied. Options:
  - First Screening: if you select this option, the screening of the bottom-most object of the PDF will be applied.
  - Most Common Screening: if you select this option, the most frequently used screening in the processed PDF document will be applied.
- **Only Unscreened Objects**: if you select this checkbox, the screening will only be applied to the objects that do not contain any screening.
- **Custom Screening Parameters**: here you can define a variable that was created earlier in the workflow (for example in the **Script** node). The variable needs to be a JSON object containing the custom parameters.

### **Example 1**

```
var theSepList = {"Cyan": { "angle": 50, "dotshape": "XXRnd", "frequency":
150 }, "Magenta" : { "angle": 60 } }
setResults({
    variables : {
        theCustomScreen:theSepList
        }
    }
);
```

The above script results in the following JSON object:

```
{"Cyan": { "angle": 50, "dotshape": "XXRnd", "frequency": 150 },
"Magenta" : { "angle": 60 } }
```

Consequently, the screening (this is, angle, dotshape and LPI) for cyan is adjusted.

#### Example 2

```
var theSepList = {"Cyan": { "angle": 50 } }
setResults({
    variables : {
        theCustomScreen:theSepList
        }
    }
);
```

The above script results in the following JSON object:

```
{"Cyan": { "angle": 50 } }
```

Consequently, only the angle for cyan is adjusted.

• Set Creator: here you can select the PDF creator, which is a PDF asset that is written inside the PDF document, referring to the application that generated the PDF. Changing the creator can be useful as some 3rd party applications ignore the metadata unless the creator has the proper value.

- **Incremental Save**: if you select this checkbox, previous sessions can be restored to a document allowing you to return to a previous version of the PDF file. If you leave this checkbox unselected, no incremental save is done which means the file size of the PDF is smaller but a previous version of the PDF cannot be restored.
- Save Edit History: if you leave this checkbox unselected, the file's history, shown in the Edit History palette in PACKZ, will be removed when saving the file.

## Set Distortion

- Horizontal Percentage: here you can select the percentage of horizontal *distortion*.
- Vertical Percentage: here you can select the percentage of vertical distortion.
- **Postfix**: here you can specify a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.

## **Replace Station**

With this node you can reload the selected stations of a tabular or nested step and repeat and replaces them with a new design.

It has the following configuration options:

- Input Step and Repeat: here you can select the step and repeat file.
- Station File: here you can define the station file that should be replaced. Select the pencil to open Expression Builder.
- Expand Repetition: if you select this checkbox, the repetitions will expand into separate, independent stations, where each station is considered as a step and repeat block. This means that the repetition will be preserved as much as possible.
- **Transform**: here you can select a transformation. Options:
  - None: if you select this option, no transformation will take place.
  - Create Verso: if you select this option, a new document (filename\_verso.pdf) will be created in which the repetition is horizontally mirrored. This is also called Turn. The stations of the step and repeat are all placeholders.
  - **Create Tumble**: if you select this option, a new document (**filename\_tumble.pdf**) will be created in which the repetition is vertically mirrored. The stations of the step and repeat are all placeholders.
  - Create Rotated 180°: if you select this option, a new document (filename\_Rotated 180.pdf) will be created in which the repetition is rotated 180°. The stations of the step and repeat are all placeholders.
- **Page Index**: here you can define a specific page number that will be selected in case a the reloaded station is a multipage PDF. By default, the first page is selected.
- **Replace Type**: here you can select which stations should be replaced. Options:
  - None: if you select this option, none of the stations will be replaced.
  - **Block Index**: if you select this option, the stations with a specific station number will be replaced. You can define the station number in the field **Block Index**.
  - **Placeholder Name**: if you select this option, the stations with a specific placeholder name will be replaced. You can define the placeholder name in the field **Placeholder Name**.
  - **Placeholder Subset**: if you select this option, the subset of a specific placeholder will be replaced. You can define the placeholder name in the field **Placeholder Name**.
  - Subset Range: here you can define the subset range in case you have selected a placeholder subset to be replaced.
  - All Empty Placeholders: if you select this option, all empty placeholders will be replaced.
  - All Stations: if you select this option, all stations will be replaced.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### Save to Device

With this node you can save a copy of the document with specific PDF options applied.

It has the following configuration options:

- Input File: here you can select the input file. Select the pencil to open Expression Builder.
- Downsample Image: if you select this checkbox, the images will be downsampled to a predefined resolution.

**Note:** Downsampling images can significantly reduce the size of a PDF as it reduces the number of pixels in an image. However, this also means the quality of the image will decrease.

• Allow Upsampling: if you select this checkbox, images with a resolution smaller than the target resolution will be resampled. Images will be downsampled to the denominator value closest to the target resolution. The value must be an integer.

## Example

•

When setting a target resolution of 72 ppi, an image of 1200 ppi will be downsampled to 75 ppi as 75 is a denominator of 1200 ppi (1200 / 75 = 16).

**Note:** Images that are part of a model or a station will be downsampled as well.

- **Image Compression**: here you can select one of the compression modes from the drop-down. When you select **Lossless**, no compression will be applied. When you select **JPEG Low Quality**, the compression rate is high which will result in low quality images but small file sizes. **JPEG High Quality** will have a low compression rate with less quality loss but the file size will be bigger.
- **Rasterize**: if you select this checkbox, the complete PDF content will be rasterized to an image with a predefined resolution.
- Separated: here you can select to output a separated PDF. Options:
  - Not: if you select this option, separation will not be applied.
  - Separation in separate files: if you select this option, each separation will be a separate PDF file.
  - Separation on new page: if you select this option, each separation will be a page of a multipage PDF.
  - Separated + composite on first page: if you select this option, each separation will be a page of a multipage PDF, and the first page contains a composite.
  - Black / White: if you select this checkbox, the separated PDF will be generated in grayscale instead of the separation color.
- **Rotated**: here you can select the transformation options:
  - Not: if you select this option, no rotation will be applied.
  - Flip Horizontal: if you select this option, the document will be flipped horizontally.
  - Flip Vertical: if you select this option, the document will be flipped vertically.
  - Rotate Left: if you select this option, the document will be rotated to the left.
  - Rotate Right: if you select this option, the document will be rotated to the right.
  - Rotate 180: if you select this option, the document will be rotated 180 degrees.
- **Color Output**: here you can select the color output. Options:
- **Composite**: if you select this option, the document will be output to CMYK + spot colors.
- **CMYK**: if you select this option, the document will be output to CMYK.
- **RGB**: if you select this option, the document will be output to GRB.
- Gray: if you select this option, the document will be output to Gray colors.
- Apply Curves: here you can define if the chosen curve preset will be applied on the entire content of the active PDF document.
  - **Note:** If you cmd+alt+click the menu option, you can use Expression Builder to specify the name of a screening from the saved screening list in the shared folder.
- Flatten: if you select this checkbox, the document will be flattened. As a consequence, the PDF will not contain any transparencies anymore. Flatten will merge all artwork in a layer, named Unlayered, and remove all empty remaining layers.

- Flatten overprints: PostScript overprints are generally supported in PDF document viewers, rips and workflows. If you select this checkbox, overprints are always flattened. If you leave this checkbox unselected, overprints may still be flattened when necessary during the process.
- **Simplify File**: if you select this checkbox, only one resource block will be written in case an image appears multiple times within the PDF document.
- Clip Images: if you select this checkbox, images will be clipped.
- Vectorize Text: if you select this checkbox, all text is converted to outlines. As a consequence, the text is no longer text and is no longer editable. All characters of the same text group become a group.
- **Remove Empty Objects**: if you select this checkbox, objects that do not contain any fill and stroke paint will be removed.
- **Remove Nested Layers**: if you select this checkbox, output layers that are nested inside other layers will be removed. Files containing this kind of structures may cause a failure or refinement in **Prinergy** (Kodak).
- **Preserve Metadata**: if you select this checkbox, the metadata of the document are preserved.
- Force Opaque Inks to Knockout: if you select this checkbox, objects in opaque inks that are in overprint will be set to knockout.

# Separations

- All: if you select this option, all separations will be outputted.
- **Process and Spot Separations**: when you select/deselect this option, the process and spot separations will be disabled/enabled.
- Technical Separations: when you select/deselect this option, the technical separations will be disabled/enabled.
- Varnish Separations: when you select/deselect this option, the varnish separations will be disabled/enabled.
- **Include Separation Names**: here you can enter the separation names that will be exported. You can use a comma separated list. If you leave this field empty, all separations will be exported.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

## Search and Select

With this node you can select objects in the PDF document that match predefined search filters defined.

The node can be combined with other **PACKZflow** nodes to apply modifications on selected objects and not on the entire PDF document.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Initial Action: here you can specify the type of selection. Options:
  - None: if you select this option, no object will be selected.
  - Select All: if you select this option, all objects matching the search criteria will be selected.
  - **Deselect All**: if you select this option, all objects matching the search criteria will be deselected, in case they were selected in a previous node (for example Layers).
  - **Inverse Selection**: if you select this option, all selected objects matching the search criteria will be deselected, while all deselected objects matching the search criteria will be selected.

## Fill

Here you can select all objects with characteristics related to fill paint mode.

• Paint: here you can select the fill paint type of the objects to select. The options are **No paint**, **Flat paint**, **Image**, **Shading** and **Pattern**. The button options can be combined and work as an AND-function.

- Color Space: here you can define the fill color space of the objects to select from a drop-down list: Separated, Lab, Gray, RGB and Registration.
- Separation Name: here you can define a separation name of the fill paint that the objects to select should contain. Additionally, you can enter alphanumeric characters to be able to use the options Begins with, Ends with, Contains and Matches.
- **Overprint**: if you select this checkbox, only the objects with a fill paint in overprint will be selected.

# Stroke

Here you can select all objects with characteristics related to stroke paint mode.

- Paint: here you can select the stroke paint type of the objects to select. The options are **No paint**, **Flat paint**, **Image**, **Shading** and **Pattern**. The button options can be combined and work as an AND-function.
- Color Space: here you can define the stroke color space of the objects to select from a drop-down list: Separated, Lab, Gray, RGB and Registration.
- Separation Name: here you can define a separation name of the stroke paint that the objects to select should contain. Additionally, you can enter alphanumeric characters to be able to use the options Begins with, Ends with, Contains and Matches.
- **Overprint**: if you select this checkbox, only the objects with a stroke paint in overprint will be selected.
- Width: here you can define a filter for searching on thicknesses of a stroke. Additionally, you can use the dropdown list to define a width that is At least, Exactly, At most or Between the entered value(s).

# Transparency

- **Object with transparency**: if you select this checkbox, only objects containing transparency will be selected.
- Blend mode: here you can specify a *blend mode* filter from a drop-down list: Any( except normal), Normal, Multiply, Screen, Overlay, Darken, Lighten, Color Dodge, Color Burn, Hard Light, Soft Light, Difference, Exclusion, Hue, Saturation, Color, Luminosity and Non Separable.
- Fill Opacity: here you can enter a percentage to define a filter for the fill opacity. Additionally, you can use the drop-down list to define an opacity percentage that is At least, Exactly, At most or Between the entered value(s).
- Stroke Opacity: here you can enter a percentage to define a filter for the stroke opacity. Additionally, you can use the drop-down list to define an opacity percentage that is At least, Exactly, At most or Between the entered value(s).
- Objects with a soft mask: if you select this checkbox, only objects containing a soft mask will be selected.

## Font

- Name: here you can enter text to define a filter for searching fonts. Additionally, you can fine-tune the filter by selecting one of the options in the drop down list: Begins with, Ends with, Contains and Matches.
- Size: here you can enter a value to define a filter for searching on font sizes. Additionally, you can use the dropdown list to define a size that is At least, Exactly, At most or Between the entered value(s).
  - **Note:** The actual font size is taken into account. In other words, when a font has been transformed, then the font size may have changed. This size will be used by the **Search and Select** node.
  - Attention: Be aware that the entire text block is selected. It may contain characters written with a font name or having a size that is not matching the search criteria.

## Objects

- Type: here you can select the type of object. Options:
  - **Barcodes**: if you select this option, all barcode objects that are recognized as being a barcode in the Assets palette will be selected.
  - Traps: if you select this option, all objects that are marked as a trap will be selected.
  - Live Objects: if you select this option, all objects listed as Live Objects palette will be selected.
  - Station Numbers: if you select this option, all objects defined as station number will be selected.
  - Clipping Paths: if you select this option, all clipping paths will be selected.
  - Station Trim Path: if you select this option, all contours that are marked with the option Mark Selected Path as Station Trim Path will be selected.
- Open Paths: if you select this checkbox, all open paths regardless the fill or stroke paint will be selected.
- **Paths Smaller than**: here you can specify a value; all objects with bounding box smaller than the defined value will be selected.
- Name: here you can enter text to define a filter for searching Object Names. Additionally, you can use a dropdown to fine-tune the filter by selecting one of the options in the drop-down list: Begins with, Ends with, Contains or Matches.
- Unscreened Object: if you select this checkbox, only objects that do not contain any screening will be selected.
- Selection Mode: here you can select a Selection Modes button that can be used to execute the selection filters.
- **Report to variable**: here you can define the name of the variable that will be created. The variable will contain information (type, location...) of the objects that are found in the document, based on the selected search criteria.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

## Simplify

With this node you can reduce the complexity of a document while preserving the preview. This is sometimes needed to reduce the file size or to optimize the file before printing.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Selection Only: if you select this checkbox, the actions will only be performed on the items that were selected in a previous node (for example Layers).

## Vectorize Images

- Vectorize Opacity: here you can create a path based on the opacity value in an opacity or alpha channel of an image.
  - Minimum: if you select this checkbox you can define the minimum opacity value.
  - Maximum: if you select this checkbox you can define the maximum opacity value.
    - **Note:** If you select both **Minimum** and **Maximum** checkboxes, two paths will be created: one based on the minimum opacity value set and one based on the maximum opacity value set.
- Vectorize Images
  - **Bitmaps**: if you select this checkbox, bitmap images will be converted to outline information, where a path will be created on the edge between the white and the black area.
  - **Minimum Intensity**: if you select this checkbox you can specify the minimum intensity of the images areas. Areas with less intensity than the one you specified are considered white, while areas with a higher value of intensity than the one you specified are considered non-white. The edge of these two areas will result in the creation of the path.

- **Smoothening**: here you can enable smoothening in order to create a smoother path during vectorization of the opacities and / or images. Options:
  - **Disabled**: if you select this option, no smoothening will be performed.
  - Average: if you select this option, the path will be created as accurate as possible in between the points.
  - Inside: if you select this option, the path will be smoothened to the inside.
  - **Outside**: if you select this option, the path will be smoothened to the outside.
- Tolerance: here you can specify the maximum divergence from the original path.

## Reduce

- **Remove pure invisible objects**: if you select this checkbox, all objects that do not contain any fill paint or stroke paint or that are completely transparent, will be removed.
- **Remove white paint on white background**: if you select this checkbox, all objects that contain a white fill and white or no stroke, positioned on a white background, will be removed.
- **Reduce Objects**: if you select this checkbox, objects that are (partially) hidden behind an object will be cut away on the intersection, except if they contain transparency. This option allows you to remove a significant amount of unnecessary content.
- Enable clipping: if you select this checkbox, clipping is enabled. If you do not enable this checkbox, this will have an influence on the **Reduce objects**, since objects can not be partially reduced in that case.
- Unite objects: if you select this checkbox, objects containing the same paint will be united.
- **Respect Groups**: if you select this checkbox, this will have an influence on the **Unite objects** function, since two objects from different groups will never be united.

### Linework

- Vectorize to Linework
  - Patterns: if you select this checkbox, all patterns will be vectorized to linework.
  - **Strokes**: if you select this checkbox, all strokes will be vectorized to linework.
  - Text: if you select this checkbox, all text will be vectorized to linework.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### **Tabular Step and Repeat**

With this node you can steps and repeats labels, flexible packaging or any PDF design in such a way that the complete printing substrate is covered and mass production is feasible and profitable.

It has the following configuration options:

• Input File: here you can define the input file. Select the pencil to open Expression Builder.

### Plate

- Width: here you can define a custom plate width.
- **Height**: here you can define a custom plate height.

### Paper

- Width: here you can define a custom paper width.
- Height: here you can define a custom paper height.

- **Reference Point**: with the reference point you can define the position of the paper on the plate, taking the horizontal and/or vertical paper offset values into account. The horizontal and vertical paper offset values can be defined in the fields **Horizontal Offset** and **Vertical Offset**.
- Horizontal Offset: here you can define the horizontal offset that is taken into account when selecting a reference point for placing the paper on the plate.
- Vertical Offset: here you can define the vertical offset that is taken into account when selecting a reference point for placing the paper on the plate.

# Step And Repeat

- **Reference Point**: with the reference point you can define the position of the complete repetition on the paper, taking the horizontal and/or vertical offset values into account. The horizontal and vertical paper offset values can be defined in the fields **Horizontal Offset** and **Vertical Offset**.
- Horizontal Offset: here you can define the horizontal offset when defining a reference point for the repetition on the paper.
- Vertical Offset: here you can define the vertical offset when defining a reference point for the repetition on the paper.
- **Make Placeholder**: if you select this checkbox, an empty placeholder will be created so that the step and repeat file can already be created, even though the PDF or AI file is not available yet.

## Repetition

- **Page Number**: in case a multipage PDF is loaded as a station, by default the first page is selected. To change this, you can define a different page by entering a page number.
- Trim Path: here you can define the area of the station that will be used to create the repetition.
- Orientation: here you can define the orientation of the station (up, down, left, right).
- Horizontal Count: here you can specify the number of horizontal stations.
- Vertical Count: here you can specify the number of vertical stations.
- Horizontal Offset: here you can specify the horizontal position of a repetition block according to the previously defined one.
- Vertical Offset: here you can specify the vertical position of a repetition block according to the previously defined one.
- Horizontal Gap: here you can specify the horizontal white gap between the stations of a step and repeat block.
- Vertical Gap: here you can specify the vertical white gap between the stations of a step and repeat block.

## Paper

- **Direction**: here you can define the staggering direction. Options:
  - No staggering
  - Vertical staggering
  - Horizontal staggering
- Vertical/Horizontal offset (depending of vertical/horizontal staggering): here you can define the vertical/ horizontal staggering distance, which will move columns upwards or downwards/left or right within a step and repeat block.
- Vertical/Horizontal Offset As Percentage (depending of vertical/horizontal staggering): additionally, the Vertical stagger offset as a percentage input field will be updated with the distance in percentages. Negative values result in downwards/leftwards movements.
- **Restart After**: if you select this checkbox, staggering will be restarted after the defined amount of columns/rows. If you leave this checkbox unselected, each column/row will be shifted accordingly to the previous column/row.
- Add Extra Station To ... columns/rows (depending of vertical/horizontal staggering)
  - Odd: if you select this option, an extra label will be added to the odd column/row.
  - Even: if you select this option, an extra label will be added to the even column/row.
- **Include partial one ups**: if you select this checkbox, partial labels will be added to the Step and Repeat to avoid ink coverage instabilities. Partial labels will not get a station number.

• Head Turn: here you can define the rotation of the labels.

### Station numbers

- Generate Station Numbers: if you select this checkbox, station numbers will be generated, which means that a unique number is placed on every station of a step and repeat block. This allows to easily recognize an error on a specific station of a printed step and repeat.
- **Prefix**: here you can define a prefix to the station numbers.
- Style: here you can define whether the station numbers should be written as numbers or letters.
- Numbering Origin: here you can define where the numbering should start.
- **Flow**: here you can define in which direction the numbering should go.

### Bleed

- Bleed: here you can define the amount of *bleed* that should be added to the step and repeat.
- Get bleed distance from station: if you select this checkbox, the bleed distance as defined in the station will be taken which will overwrite the value set in the **Bleed** input field.
- Mode: here you can select the bleed mode. Options:
  - Center bleed between adjacent stations: if you select this option, the bleed for the selected step and repeat block will be calculated taking the entire step and repeat into account.
  - **Ignore surrounding blocks when calculating bleed**: if you select this option, the surrounding blocks will be ignored when calculating bleed.
- Left bleed limit for this block: here you can specify a left bleed limit. This value must be smaller than or equal then the bleed value.
- **Right bleed limit for this block**: here you can specify a right bleed limit. This value must be smaller than or equal then the bleed value.
- **Top bleed limit for this block**: here you can specify a top bleed limit. This value must be smaller than or equal then the bleed value.
- **Bottom bleed limit for this block**: here you can specify a bottom bleed limit. This value must be smaller than or equal then the bleed value.



**Note:** The bleed value will not be added to the vertical and horizontal step of the **Repetition** parameters. Consequently, bleed in between labels will not be visible if there are no gaps defined.

- Seamless: if you select one of the Seamless options, seamless step and repeats for cylinders will be created.
  - **Horizontal Seamless**: if you select the Horizontal Seamless option, the horizontal paper size will be filled with as much labels as possible, equally dividing the white spaces between the labels.
  - Vertical Seamless: if you select the Vertical Seamless option, the vertical paper size will be filled with as much labels as possible, equally dividing the white spaces between the labels.
- **OneUp File List**: here you can define a variable defining extra 1up files that should go to subsequent blocks. The variable consist of a JSON formatted list.
- **OneUp File List Parameters**: here you can define a variable corresponding to each 1up defined in the above file list. The variable consists of a JSON list of parameter objects. Each item corresponds to an item at the same position in the above **OneUp File List**.

### Example of a workflow that will create an multi block Tabular Step And Repeat job

The workflow consists of 4 nodes:

## 1. Start From Kiosk

2. Script: this node will setup the variables theFileSet and theFileParams to be used by the next Tabular Step and Repeat node. The script contents are as follows:

```
var theFile1 = "cloudflow://PP_FILE_STORE/CFTest/TabSnR/A.pdf";
var theFile2 = "cloudflow://PP_FILE_STORE/CFTest/TabSnR/B.pdf";
var theFiles = [theFile1, theFile2];
var theParams = [ {"VCount":3}, {"Orientation":"left", "VCount":2} ];
setResults({
```

```
variables:{
   theFileSet:theFiles,
   theFileParams:theParams
   }
};
```

- 3. Tabular Step and Repeat: this node will do the actual work if the needed parameters are configured this way:
  - a. Input File: the incoming file which was submitted to the workflow
  - b. OneUp File List: variable theFileSet
  - c. OneUp File List Parameters: variable theFileParams
- 4. End

The result will be a step and repeat file containing 3 blocks:

- 1. A first block with a Step and Repeat defined in the UI parameters of the **Tabular Step and Repeat** node. The lup file is the incoming file which was submitted to the workflow.
- 2. A second block with a Step and Repeat done on file **A.pdf**. In this block the parameters are the same as in the first block except the Vertical Repeat count is set to 3.
- **3.** A third block with a Step and Repeat done on file **B.pdf**. In this block the parameters are copied from the second block, with an extra override of two values: **VCount**" (vertical count) set to 2 and **Orientation** set to left.
- Start Applying Variables From First Block: if you select this checkbox, the variables defined in OneUp File List and OneUp File List Parameters will be applied from the first block.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.
- **Note:** PACKZ can be used to inspect the result and make changes if necessary. See the PACKZ Reference Manual for more information.

# Text

With this node you can perform search and replace operations on text.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Selection Only: if you select this checkbox, the search area will be restricted to a selected region that was defined in an earlier node (for example Layers).
- Find: here you can specify a search term.
- Hits Must: here you can enter alphanumeric characters to be able to use the options Begins with, Ends with, Contains and Matches.
- Match Case: if you select this checkbox, the search term's case will be taken into account in the result.
- Error if nothing found: if you select this checkbox, the flow will generate an error when the specified search term is not found in the PDF.
- Replace With: here you can specify a term that will replace the search term.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

# Transform

With this node you can move, rotate, reflect or scale selected objects or the entire PDF document.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Selection Only: if you select this checkbox, the actions will only be performed on the items that were selected in a previous node (for example Layers).
- **Reference**: here you can select the reference according to which the transformation should be done. It can be a *pagebox* or a selected object's bounding box.
- Fit to reference box: if you select this checkbox, the selection is fit to the selected reference box. If no object is selected, the complete document will be fit to the reference box.
- Reference Point: here you can select the transformation position of the selection's bounding box reference point.
- Horizontal Distance: here you can define the horizontal moving distance of an object. Negative values will move the object to the left. Positive values will move the object to the right.
- Vertical Distance: here you can define the vertical moving distance of an object. Negative values will move the object to the bottom. Positive values will move the object to the top.
- Horizontal Scale: here you can define the horizontal scaling distance of an object. The input fields range from -100% to +100%. A negative value will flip the bounding box.
- Vertical Scale: here you can define the vertical scaling distance of an object. The input fields range from -100% to +100%. A negative value will flip the bounding box.
- Mirror: here you can select if an object should be mirrored. Options:
  - None: if you select this option, no mirroring will be performed.
  - Horizontal: if you select this option, the selection will be mirrored horizontally.
  - Vertical: if you select this option, the selection will be mirrored vertically.
- **Rotate**: here you can define the rotation value of a selection. If you use a negative value, the bounding box will be rotated clockwise; if you use a positive value, the bounding bow will be rotated counterclockwise.
- Delete Selected Objects: if you select this checkbox, all selected objects will be deleted.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

**Note:** the **Search and Select** node is very often combined with the **Transform** node.

## Trap

With this node you can automatically create *trapping* between abutting colored objects within a document.

It has the following configuration options:

- Input File: here you can define the input file. Select the pencil to open Expression Builder.
- Trap On Selection Only: if you select this checkbox, only the selected objects will be trapped.

# Distance

- **Default**: here you can define the default trapping distance that is used to create a trap.
- **Into Black**: here you can define the distance for trapping between any paint and black. This trap distance is typically used for linework using black keyline contours, like in comics. When the trap distance is bigger than the black keyline contour, then the trap will be centered under the keyline.
- Into Spot: here you can define the distance for trapping between any paint and spot colors.
- Into Opaque: here you can define the distance for trapping between any paint and opaque colors.
- Into Image: here you can define the distance for trapping between any paint and images.
- **Overlap**: here you can define the trapping overlap, which allows a slight spread in the direction opposite to the trap direction so that small gaps are eliminated. This can be useful since a trap is per definition a spread object

that follows the contour of the object on which it is applied. This is done with specific precision and may cause extreme small gaps. Those artifacts may become visible on high-resolution image setters and may cause a visual artifact on the printed result.

- Intensity: here you can define the intensity of the trap object. The trap object is a new object containing the paint of the trapped object with a *Blend Mode* Darken applied. The default intensity used is 100%. However, an intensity of 100% can result in visible traps when the color difference between the two touching objects is too high. Reducing the intensity of the trap object will solve the problem.
- **Reverse Trapping**: if you select this checkbox, reverse trapping will be applied. Reverse trapping is used in the metal printing industry or in Dry Offset where fluid inks are used and cause a natural trap when the inks coalesce. By generating a white trap, an overlap of opaque inks can be prevented.
- Corners: here you can select the corner of the trapping.
  - Beveled corner
  - Rounded corner
  - Mitered corner
  - Square corner

# Pull Backs

- Type: here you can specify the *pull back* behavior. Options:
  - All: if you select this option, the lightest separation will be pulled back from the edge of a rich black or rich color.
  - Only Black: if you select this option, the lightest separation will be pulled back from the edge of a rich black
  - **Only Flats**: if you select this option, a pull back is performed only when the resulting trap would have a flat paint.
  - None: if you select this option, no pull back will be performed.
- **Pull Back Type**: here you can select the pull back type. Options:
  - Single: if you select this option, pull backs are only allowed in case only one single separation is pulled back.
  - All But One: if you select this option, pull backs are only allowed in case only all but one separation is pulled back.
  - Both: if you select this option, pull backs are always allowed.
- **Distance**: here you can select the distance for pulling back the lightest separation from the edge of a rich black or rich color.
- **Maximum Color Difference**: here you can define the maximum color difference for between the two touching objects.

# **Trap Direction**

- Minimum Ink Difference: here you can define the minimum ink difference.
- Maximum Luminosity Difference: here you can define the maximum luminosity difference.
- Image Trapping: here you can define the trap direction for any paint to images. Options:
  - Automatic: if you select this option, the trapper calculates the trap direction automatically and can spread or choke the image.
  - Always Towards: if you select this option, a trap towards the image is applied.
  - Always Away: if you select this option, a trap away from the image is applied.
  - **Note:** Always Towards or Always Away from image allows applying a trap that is going in the same direction no matter which paint it is touching. When choosing one of these options, the trapper will ignore equal paints on top.

- Shading Trapping: here you can define the trap direction for any paint to shadings. Options:
  - Automatic: if you select this option, the trapper calculates the trap direction automatically and can spread or choke the shading.
  - Always Towards: if you select this option, a trap towards the shading is applied.
  - Always Away: if you select this option, a trap away from the shading is applied.
  - **Note:** Always Towards or Always Away from shading allows applying a trap that is going in the same direction no matter which paint it is touching. When choosing one of these options, the trapper will ignore equal paints on top.
- Towards Opaque Separations: if you select this checkbox, a spread towards opaque inks will always be generated.

### Processing

• **Ignore gaps smaller than**: here you can define the maximum size that small gap between touching objects can contain to be ignored by the trapper.

### Vectorizing

- Vectorize Bitmaps: if you select this checkbox, bitmap images will be converted to outline information, where a path will be created on the edge between the white and the black area.
- Vectorize Images: if you select this checkbox, images will be vectorized.
- Vectorize Opacity: if you select this checkbox, a path based on the Minimum Opacity and Maximum Opacity value sets will be created.
- **Minimum Opacity**: here you can define the minimum opacity that will be taken into account when vectorizing opacity.
- Maximum Opacity: here you can define the maximum opacity that will be taken into account when vectorizing opacity.
- **Smoothen**: here you can enable smoothening in order to create a smoother path during vectorization of the opacities and/or images. Options:
  - **Disabled**: if you select this option, no smoothening will be performed.
  - Average: if you select this option, the path will be created as accurate as possible in between the points.
  - Inside: if you select this option, the path will be smoothened to the inside.
  - **Outside**: if you select this option, the path will be smoothened to the outside.
- Smoothen Tolerance: here you can specify the maximum divergence from the original path.

## **Existing Traps**

- Existing Traps: here you can select what to do when traps are present in the design that were made with PACKZ, and new traps are automatically added to a selection or the entire job. Options:
  - Keep: if you select this option, existing traps are not replaced.
  - Replace: if you select this option, existing traps are replaced
- Unknown Traps: here you can select what to do with incorrect or irrelevant traps that may have been applied on transformed or removed objects. Options:
  - Ignore: if you select this option, the trap stays unmodified.
  - **Remove**: if you select this option, the trap is removed or recalculated.

### **Clip Trap Images**

• Clip Trap Images: if you select this checkbox, the image paints to the bounding box of the traps are clipped after trapping. This is done to handle problems on RIPs that load the entire image for every reference to the same image.

- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

**Note:** PACKZ can be used to manually alter the traps.

### Variable Data Printing

With this node you can execute VDP.

It has the following configuration options:

- Input file: here you can define the VDP input file that was prepared in PACKZ and linked with variable elements. This can be a 1up file or a Step&Repeat file. Select the pencil to open Expression Builder.
  - **Note:** See the PACKZ manual for more information about Prepare VDP.

### **Database Settings**

- Override CSV file: here you can define a CSV file in case you want to override the CSV file that is linked to the single design.
- Assets folder: here you can define a folder that contains resources that are used in VDP. For example external PDF files, external images...

### **Divide Parameters**

- Start from Item: here you can define from which record onwards you want to generate output files.
- **Number of Output Items**: here you can define the number of records you want to output. If the number is higher than the number of items (records in the database), blank pages/documents will be generated.
- Create: here you can define the type of file you want to create. Options:
  - Single File: if you select this option, each record will exist as a separate page in a multipage PDF.
  - Multiple Files: if you select this option, each record will be output as a single PDF file.
  - **Multipage Multiple Files**: if you select this option, multiple multipage files are generated, based on the number you enter in the parameter **Split document after page count**.
  - Single File Part: if you select this option, the output result will be similar to the result in case you select Single File, but the output is divided into parts, which allows you to perform parallel processing. See Calculate only on page 215 for more information. Make sure to use the Join Pages node later to join the PDF files again.
    - **Divide Document Into Parts**: here you can define in how many parts the output file will be divided? For efficiency reasons, this value should be the number of PACKZFLOW workers that are available for VDP execution.

### Example

There are 4 PACKZFLOW workers available. The **Repeat** node creates 4 workables (this can be variable) and the output file is divided into 4 parts. This means that every workable will do an execution task. Consequently, for a 1up file with 400 records, every workable will do execution of 100 records.

• Select part index to output: here you can define the part index. This will be the variable repeat.index, since every workable needs to have a different task, they cannot output the same records.

See Partitioning VDP on page 215 for an example and more detailed information on this topic.

### Settings

• **Calculate Only**: if you select this checkbox, the Variable Data Printing node will not execute the VDP, but you will get information about the output VDP. The node will create a variable that contains information about

the number of items in the database and number of pages that are going to be output for the full database. See Calculate only on page 215 for more detailed information about this parameter.

- **Output Results to Variable Name**: here you can define the variable name that will output the values that you can use to create your parallel execution.
- Max Concurrent Workers: here you can define a maximum number of concurrent workers that are used to output the VDP file.

## **Fill Parameters**

- **Output Method**: here you can select how the VDP items are organized on the output file. Options:
  - From file: if you select this option, the node will use the parameters that you defined in the PDF file in Prepare VDP window in Packz.
  - Sheet fed: if you select this option, the output is organized by sheets called frames.
    - Start: here you can define the start of the sheets. Options:
      - Top to Bottom: if you select this option, the first printed sheet is the start of the database records. This means that the sheets that go on top will have higher numbers. The top one will have the highest record number.
      - Bottom to Top: if you select this option, the first printed frame will contain the last records. Consequently, the top of the printed stack will contain the first record.
    - Item Start: here you can define the corner of the frame where the first record will be placed.
    - Item Filling: here you can define the flow direction of the subsequent records.
    - **Priority**: here you can define the fill priority. Options:
      - **Horizontal**: if you select this option, the subsequent records will first fill one frame, and then the next frame.
      - Vertical: if you select this option, the subsequent records run throughout all frames in one vertical column.
  - Web fed: if you select this option, the output is organized by blocks, a continuous roll containing a selected number of rows.
    - Block size: here you can define the size of one block.
      - **Note:** This number does not have to be equal to the number of vertical repetitions in your Step&Repeat page. A block is seen independently from your current page. A block can contain multiple pages.

Options:

- **Complete Job**: if you select this option, the number of rows is automatically determined by the number of records in the database.
- **Rows**: if you select this option, you can define the number of rows in one block in the number field on the right.
  - Gap between Blocks: here you can define the number of empty rows between the blocks.
  - Start: here you can define the start of the blocks. Options:
    - Top to Bottom: if you select this option, the first printed block is the start of the database records. This means that the blocks that go on top will have higher numbers. The top one will have the highest record number.
    - Bottom to Top: if you select this option, the first printed block will contain the last records. Consequently, the top of the printed roll will contain the first record.
- Item Start: here you can define the corner of the block where the first record will be placed.
- Item Filling: here you can define the flow direction of the subsequent records.
- **Output**: here you can define the output path. Select the pencil to open Expression Builder.

### Calculate only

With this option you can create a variable to split your workflow into multiple sub-workables.

If you select the checkbox **Calculate Only** in the Variable Printing Data node, the node will not execute the VDP, but it will create a variable that contains information how you can split your workflow into multiple sub-workables to run PACKZflow VDP in parallel.

In the parameter **Output Results to Variable Name**, enter the variable name that will output the values that you can use to create your parallel execution. For example, in case of a 4x3 repetition input file, the variable output will have the following contents:

```
ranges: [
{
    count: 540,
    start: 1
}]
total: 540,
details: {
   DatabaseSize: 21,
   VDPStations: 54,
   NonVDPStations: 0,
   LiveObjectsOutsideVDP: 0,
   LiveObjectsInDynamicContent: 108,
   RegularObjectsOutsideVDP: 0,
   RepeatColumn: " N ",
   RegularObjectsInDynamicVDP: 0,
   RegularObjectsInStaticVDP: 270,
   LiveObjectsInStaticContent: 0,
    DatabaseSizeWithRepeatsOptional: 540
}
}
```

- DatabaseSize: 21 shows the number of records are in the database.
- **DatabaseSizeWithRepeatsOptional: 540** shows the number of total records that will be used when repeating the records, if you use repeats in your VDP configuration. If there are no repeats in your configuration, the number will be equal to **DatabaseSize**. In the example above, the database is repeating using the record name **N**.

### **Partitioning VDP**

You can split VDP into multiple parts that can be executed in parallel by separate workers.

To activate this option, select the parameter field **Divide Parameters** > **Create** > **Single File Part** in the **Variable Data Printing** node. This will reveal two extra parameter fields:

- Divide Document Into Parts: here you can define the number of parts the job will be divided.
- Select part index to output: here you can select which part of the divided parts the node will use. The range is from 0 to the number of parts-1. The node will then execute only one part. For example, if you want to output 1.000 items and divide the output to 4 parts, set the parts index to 2. In that case, the node will output a PDF that contains items from 500 to 750. With this setting, you each of the parts of 0-250, 250-500, 500-750 and 750-1000 are generated with separate workers, which will result in 4 PDF documents. Once the PDF files are generated, use the Join Pages node to generate a single PDF.

### Example workflow:



The workflow is creating VDP output from a 1up file, creates the repetition page using Tabular Step and Repeat and splits the work into multiple parts.

## The Repeat node is a simple counter:

Repeat	?
Mode Counter v	
Create Multiple Workables	
Variable name repeat	
Nr Repeats	
numOfParts	1
	Close Save

The repetition is configured with the variable **numOfParts**, which is a value entered by user. The **Repeat** node creates multiple workables, so the node **Variable Data Printing** node is executed in parallel, subject to a number of licensed PACKZFLOW workers.

The Divide Parameters section in the Variable Data Printing node is configured the following way:

Divide Parameters	
Start from Item:	1
Number of Output Items:	0
Create:	Single File Part
Divide Document Into Parts:	numOfParts
Select part index to output:	repeat.index

- Number of Output Items is set to 0, which means that all records from the database will be output.
- Select part index to output is configured with a subsequent part from the Repeat node.

Once all parts are generated, you can use an extra **Script** node which will put all PDF output files in a correct order. The Script node has the following configuration:
Ensure Order (Script)	
Input Files:	
Folder Path from Tabular Step and Repeat Filename w/o ext. from Tabular Step and Repeat	
Variables For Script: All Variables from workable Variables	
Output connectors:	
Log function feedback	
Fail on function errors	
Max Concurrent Workers: 2	
Include scripts:	
Script File:	
<pre>1 var theFile = getParameters().files; 2 3 var theNumberOfParts = getParameters().variables.numOfParts; 4</pre>	
<pre>5 var theFiles = []; 6</pre>	
<pre>7 for( var i = 0 ; i &lt; theNumberOfParts; ++i ) 8 * {</pre>	
<pre>9 theFiles.push( theFile + i + "of" + theNumberOfParts + "_VDP.pdf" ); 10 } 11</pre>	
<pre>12 setResultFiles( theFiles );</pre>	
	0

The configuration will create a list of PDF references in the correct order.

Once the right order is generated, the Join Pages node creates a single PDF output.

F

Note: Join Pages is optimized for the output of a partitioned PDF from the VDP node, so it will process the join very quickly, even if the PDF parts contain a big number of pages.

# Warp

With this node you can map cells of a source grid to cells of a destination grid thus digitally manipulating artwork to fit printing substrates that have (or in the printing process take) a shape that is different from the artwork's original shape.

#### Attention: License required $\triangle$

Warping (except for Conical Warp) and 3D requires an extra license key. Please contact the Sales Representative for your region. For some 3D applications an IC3D license is needed.

- Input File: here you can select the input file. Select the pencil to open Expression Builder. The default is set to use the full file URL from the previous node.
- Layer Names to Warp: here you can specify one or more layers to warp. You can separate several layer names with a comma. The layer names can be specified with a wild card. For example, if you put Layer\*, all layers starting with the word Layer will be selected.
- Image Resolution: here you can specify the resolution to use for the images that are warped. Options:
  - Keep original objects: if you select this option, the original artwork will be kept after warping.
  - Allow distortion: if you select this option, the file will be printed with possible distortion applied, but only to objects.

Objects will be moved to the destination grid on the corresponding position and rotated accordingly to the shape of that position of the destination grid, without distorting the artwork itself.

- **Reverse Warp**: if you select this option, it allows warping the selection from the destination grid back to the source grid.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

### White Underprint

With this node you can automatically add a *white underprint* to artwork.

It has the following configuration options:

- Input File: here you can specify the input file. Select the pencil to open Expression Builder.
- Separation Name: here you can specify the separation to which the white underprint will be added.
- Intensity: here you can enter the percentage to set the intensity of the white object's separation color.
- **Distance**: here you can define the distance of the spread or choke value used to create the white underprint object. You can define spread or choke by activating one of the segmented buttons.
- **Corners**: here you can select the type of corner to generate the white underprint object. Options are beveled, rounded or square.
- Mode: Here you can select the type of white underprint. Options:
  - **Subtract**: if you select this option, the selected objects will be subtracted from the existing white underprint object.
  - Add: if you select this option, a unite will be applied on the selected objects, generating a new object in 100% of the defined separation in overprint. The white underprint object will be positioned in a new layer, called White Underprint.
- **Output**: here you can select the output type. Options:
  - **Postfix**: if you select this option, you can define a postfix that will be added to the newly generated file. The new file will be placed on the same location as the original file.
  - **Custom**: if you select this option, you can specify a path where the new file will be placed and a postfix that will be added to the new file. Select the pencil to open Expression Builder.

# RIP

# **CLOUDFLOW RIP**

With this node you can integrate the RIP functionality in a workflow.

It contains several categories:

- Input
- Output

- Separations
- Calibration
- Color Management
- CIP3

Note: The default settings in the RIP node renders any PDF/X-4 file correctly. In the section Color Management > Output Profiles, make sure that following options are set to their default values:

- Ignore ICC tagged objects should be OFF.
- Preserve black separation should be OFF.

See also **RIP** on page 330.

### Input

• Files to RIP: here you can select the files to RIP. Select the pencil to open Expression Builder.

### Output

• Output folder: here you can select the output folder for the RIPped files.

# **Output Type**

- **Output**: here you can define what kind of file the RIP has to produce. Options:
  - Composite: if you select this option, the RIP will generate a composite file
  - Separated: if you select this option, the RIP will generate separated files without a screen
  - Separated Halftone: if you select this option, the RIP will generate separated files with screen
- Bits per channel: here you can define the number of bits per separation. For example:
  - 1-bit tiffs =  $2^1$  gray levels = 2 = white (0) or black (1)
  - **2-bit tiffs** =  $2^2$  gray levels = 4
  - 4-bit tiffs =  $2^4$  gray levels = 16
  - 8-bit tiffs =  $2^8$  gray levels = 256
  - **10-bit tiffs** =  $2^{10}$  gray levels = 1.024
  - **16-bit tiffs** =  $2^{16}$  gray levels = 65.536

# Resolution

- Vertical: here you can define the vertical resolution of the output file.
- Horizontal: here you can define the horizontal resolution of the output file.

# **Output Properties**

- Pages: here you can define the pages that you want to output. You can select all pages or a range of pages.
- Rotation: here you can rotate the file for output.
- **Page Box**: here you can select the page box for output.
- Horizontal Distortion: here you can define the horizontal distortion.
- Vertical Distortion: here you can define the vertical *distortion*.
- Seamless screening: if you select this checkbox, the screening in the file is horizontally continuous between the jobs, from left to right. This means that, if you would put the beginning and the end of the output file together in a horizontal direction, there would not be any gaps or broken dots. Seamless screening is especially used when making flexo cylinders, for example for wall paper.
- Negative Output: if you select this checkbox, the output of the file will be negative.
- Mirror Output: if you select this checkbox, the output of the file will be mirrored.

• Enable Harlequin Precision Screening: if you select this checkbox, Harlequin Precision Screening (HPS) is enabled.

HPS is a color screening technology that ensures high-quality reproduction with any screening option in the RIP (defined using a spot function). It allows you to select any screen frequency and to use the usual CMYK screen angles of 0°, 15°, 45°, and 75° (plus multiples of 90°). To reduce moiré patterning, HPS uses an adaptive screening technique that can adjust each halftone dot so that it is placed within one half pixel of its ideal location. HPS also ensure rosettes are always hole centered, and it will generate extra gray levels, allowing the use of higher screen frequencies than the resolution would normally allow.

Enabling HPS screening sometimes results in another ruling or angle than requested. This is a consequence of the Precision Screening, which prevents rosettes from shifting. The ruling and angle are always optimal as HPS sets out to achieve the best possible match with the parameters the user defined. When you select the **Favour Angles** option, the algorithm favours angles to best match your parameters. When you select the **Favour Ruling** option, the algorithm favours ruling to best match your parameters.

# **TIFF Format**

- Strip format: here you can select if the output file should contain one or multiple strips.
- Compression: here you can select the compression of the output file.
- Anti-aliasing: here you can select the anti-aliasing type.

**Note:** Anti-aliasing is only possible in case of with 8-bit TIFF files.

- Bit order:
  - Reverse bit order: if you select this checkbox, the TIFF file will be output with reverse bit order.
  - **Pad to 32-bit alignment**: if you select this checkbox, each line of the TIFF file data will end on a multiple of 32 bits. This is an efficiency setting, for monochrome output only, that may make the file faster to read in some applications.

# Separations

# **Default Screening Settings**

- Job Metadata
  - **Retrieve halftone settings from the Job**: if you select this checkbox, the metadata of the file will be used. If there are already screens defined inside the file, the RIP will take these to calculate.
- **Default ruling**: here you can define the ruling.
- **Default dotshape**: here you can define the dotshape.
- HXM Min Dot Size: here you can define the minimum dot size (in μ) in case you are using HXM screening. See HXM screening sets on page 340 for more information on how to calculate the minimum dot size and the supported HXM resolutions.
- **Rescale images**: in some cases it is useful to upscale all images in a file before ripping to improve the occurrence of broken dots by adding extra interpolated pixels in between. Harsh pixel transition is smoothened out with

cleaner dots as a result. The drawback is that this technique will lead to a performance hit and a slight blurring of the image data. Options:

• None: if you select this option, no image rescaling is done, each pixel is screened in itself.



• Automatic: if you select this option, all images are upscaled to a higher resolution to lower the amount of broken dots.



• **Full Resolution**: if you select this option, all images are upscaled to the full ripping resolution. This gives the best results, but comes with the greatest performance cost.



- **Note:** For RIP resolutions below 2000 dpi, Automatic and Full Resolution are basically the same.
- **Low-Pass Resample**: if you select this option, the resolution of the image does not change. In stead, a filter is used on the image that causes a blur. Consequently, the color values are closer to each other and the transition of the dots is less abrupt. This comes with a image quality cost.



# **Separation Specific Settings**

- Override defaults: here you can give extra options in the separation list underneath.
  - Set ruling per separation: if you select this checkbox, you can define a specific ruling for each separation in the separation list underneath.
  - Set dotshape per separation: if you select this checkbox, you can define a specific dotshape for each separation in the separation list underneath.
- Empty separations
  - **Output empty process colors:** if you select this checkbox, a TIFF file will also be generated for process colors inside a file that don't contain information (for example in a duotone file).
  - Spot color screens: here you can define what should be done with the spot color screens. Options:
  - **Don't output spot colors**: if you select this option, the spot colors will not be output.
  - Iterate over process color screens: if you select this option, the same angles of the process colors will be used for spot colors.
  - Specify explicitly if you select this option, you can set the angles yourself in the separation list underneath.

- Angle family: here you can select the angle family. Options:
  - Choose to populate: if you select this option, you can define the angles yourself in the separation list.
  - Offset: if you select this option, the predefined offset angles will be entered in the separation list.
  - Flexo: if you select this option, the predefined flexo angles will be entered in the separation list.
    - **Note:** You can choose which separation you want to process or not by selecting/deselecting the checkboxes in the **Print** column in the separation list.

# Calibration

### **Dot Gain Curves**

- Enable Curve 1: here you can select the reference curve.
- Enable Curve 2: here you can select the compensation curve.
- Enable Curve 3: here you can select the bump curve.
- **Important:** The order of the curves is important for processing. The order is shown at the bottom of the window: Application Order: input % > 1 > 2 > 3 >Output %
- **Note:** In some cases the calibration is split up in a plate curve and a press curve. In that case: curve 2 is the press curve and curve 3 is the plate curve including the bump curve.

#### **Calibration Results**

Select the disclosure button to reveal this option.

**Separation**: here you can select a separation from the drop-down list to display its calibration results from the curve settings that are selected in the RIP.

#### **Color Management**

In Color Management you can apply a color profile to the job.

#### **Input Profiles**

- Input RGB Profile: here you can select the input RGB profile.
- Input CMYK Profile: here you can select the input CMYK profile.

#### **Output Profiles**

- **Output Profile**: here you can select the output profile.
  - **Ignore ICC tagged objects**: if you select this checkbox, objects tagged with an ICC profile will be ignored when applying a profile to the job.

#### CIP3

- Generate CIP3: if you select this checkbox, the RIP will generate a CIP3 file to send to the press.
- **Preview Resolution**: here you can define a preview resolution for the CIP3 file.
- Paper Width: here you can define the width (in mm) of the paper that will be used to print on.
  - Inverted polarity: if you enable this option, the file will be output with inverted polarity.
- Mirrored: if you enable this option, the file will be mirrored.
- Rotation: here you can define a rotation angle.
- Image Encoding: here you can define the image encoding.
- Image Compression: here you can define the type of compression.
- Device: here you can define a device for the CIP3 file.

#### **Intelligent Flexo**

With the node INTELLIGENT FLEXO you can apply screening and effects that are highly customised to your production requirements to guarantee optimal plate making and printing quality.

Intelligent Flexo		0
TIFF files: Processed files:	All files from node previous Node Folder Path Filename w/o extIF Filename ext.	0
Patterning	Overwrite existing file	
	□ Fill with pattern	
Cell wall:	0 μm	
Skip dots smaller than:	0 μm	
Pattern image:		Ø
	Close	ave

The node processes one or more incoming TIFF files and outputs these to a dedicated folder. These settings can be freely configured with the Expression Builder.

#### **Tip:** Compatibility

As the INTELLIGENT FLEXO node optimises TIFF files, its functionality can be used in conjunction with the Len To TIFF and Merge TIFF files nodes.

It has the following configuration options:

#### Input/Output Info

A

**TIFF files**: This field displays the input TIFF files that will be screened. Select the pencil to open Expression Builder.

**Processed files**: This field displays the output TIFF files that will be screened and their location. Select the pencil to open Expression Builder.

Overwrite existing file: Select this check box to overwrite previously screened files.

If this box is not checked and a new TIFF file is generated that has the same name as a previous one, the node will throw an error.

#### Patterning

A TIFF's dots can be printed either 'AS IS' or with additional screening to enhance output quality. In this section you define the surface patterning that you want to apply to the TIFF files. Non-printing patterns can be applied to specific regions of a TIFF to permanently mark customer specific plates in a non-erasable, non-printable manner (e.g. to protect mounting mark areas).

Fill with pattern: Select this check box to enable pattern screening.

**Cell wall**: This field defines the cell wall width. Giving a dot a solid cell wall prevents excess ink from spilling out and the dots from blurring.

**Skip dots smaller than**: This field defines which dots are to be ignored for pattern screening on the TIFF files. Dots that are too small are unsuitable for practical use in that they will fail to be properly exposed onto the plate, or fail to hold and transfer ink properly, or get damaged during the washing of the plate.

**Pattern image**: This field displays the file path to the 1-Bit TIFF image that you want to use for patterned screening. Select the pencil to open Expression Builder.

#### 

Patterns used can be common, specific designs already in use, or newly customized images combining the benefits of your experience and the power of HYBRID's advanced technology.

# **Dry Offset Screening**

With this node you can update a set of TIFF files for Dry Offset printing.

This means that any overprinting separations will be removed, so at any pixel there is only one separation that is printing. Optionally, you can make sure that two different printing separations do not touch each other.

The node has the following configuration options:

- TIFF Files: here you can define the TIFF files you want to process. Select the pencil to open Expression Builder.
- **Processed Files**: here you can specify the name to assign to the processed TIFF files. Options:
  - If you leave this parameter **blank**, the original files will be overwritten.
  - If you define an absolute CLOUDFLOW URL containing a filename (cloudflow: //..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you define an absolute CLOUDFLOW URL without a filename (cloudflow: //..., see Server URL for more information), an output file with the original file name will be written in the specified folder.
  - If you enter a **relative path containing a file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
  - If you enter a **relative path without a file name**, an output file with the original file name will be written in the specified folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- Keep Away Distance: here you can define the distance that should kept clear between two different separations. If you do not need to keep a distance, set the value to **0**.
- Sort Separations: here you can define how the separations should be ordered for processing. Options:
  - **Darkest Separation First**: if you select this option, the color found in the TIFF files will be used to sort the separations from dark to light, and the darkest separation will be on top.
  - File Order: if you select this option, the separations will be processed as specified in the TIFF Files parameter, which means that the first file will be on top.
  - Inverted File Order: if you select this option, the separations will be processed in the opposite order as specified in the **TIFF Files** parameter, which means that the last file will be on top.
- **Ignore Separations**: here you can specify the separation(s) you want to exclude from the dry offset screening. You can use a Regular expressions on page 475 and match more than one separation. If you do not want to exclude separations based on their name you can leave this parameter blank.

**Note:** You can use https://regex101.com as an online regex tester.

- **Ignore Technical Separations**: if you select this checkbox, all separations of type **technical** are excluded from the dry offset screening. If you leave this checkbox unselected, separations are not excluded based on their type.
- Marks Separation: here you can specify a separation that only contains marks. Other separations will not be modified (at all) where the marks separation contains on-pixels. This can be used to make sure that registration marks are not erased even though they are overprinting. If you do not have/need a special marks separation you can leave this parameter blank.
- **Keep Away Separation**: here you can specify the separation you want to use to get keepaway feedback. That is, all the pixels where in the input one or more separations were turned on and in the final result all separations are turned off by the dry offset screening will be marked in the specified separation. If you do not want/need this information, you can leave this parameter blank.
- **Note:** You can set the **Info Separation** and **Marks Separation** to the same value if you want/need.

# **Resample TIFF files**

With this node you can resample TIFF files.

The node has the following configuration options:

- Input Files: here you can define the TIFF files you want to process. Select the pencil to open Expression Builder.
- **To file or folder**: here you can specify the file or folder to copy to. If you specify a folder, then the file will be copied with the original name into that folder. If you specify a file, then the file will be copied exactly to the specified name. Select the pencil to open Expression Builder.
  - **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- Width: here you can select the image width.
- **Height**: here you can select the image height.
  - **Note: Resampling proportion**: You can define both the image width and height.
    - to increase or decrease
    - with as values accepted both units and decimals

The Resample TIFF files node allows non parallel TIFF rescaling, whereby width and height can vary disproportionately.

- Mirror: here you can select if the TIFF file should be mirrored.
- Rotate: here you can select the rotation of the TIFF files. The files can be rotated 180°, 90° CW or 90° CCW.

# Network

# Call REST

With this node you can call a function on a REST server.

It has the following configuration options:

- **HTTP Method**: here you can define the HTTP method.
- Content Type: here you can define whether to use JSON or XML to communicate with the server.
- URL: here you can define the URL of the REST server.

You can use explicit parameters. Select + to add a parameter.

- Name: here you can enter the name of the parameter. Select the pencil to open Expression Builder.
- Value: here you can enter the value of the parameter. Select the pencil to open Expression Builder.
- Request Data: here you can define the request data. Select the pencil to open Expression Builder.

**Note:** In case file upload is enabled, the request data needs to be a JSON object (or a serialized JSON object) that only contains string values.

- Upload files: here you can define if files needs to be uploaded or not. Options:
  - None: if you select this option, no file will be uploaded.
  - Single: if you select this option, only one file will be uploaded.
  - **Multiple**: if you select this option, no file will be uploaded, the REST call is executed for every file, resulting in an array of the responses.
- Field name for uploaded file: here you can define the name of the field the uploaded file is associated with. You can change this if it is required by the REST server.
- File to upload: here you can select the file(s) to upload. Select the pencil to open Expression Builder.
- Verify SSL Certificate: if you select this checkbox, the certificate of the server is verified (HTTPS only). Disable this checkbox when using self-signed certificates.
- SSL Version: here you can define the SSL version to use (HTTPS only). In normal cases you can set the value to Automatic, but some servers require to specify the SSL version.
- Authentication: here you can specify the HTTP authentication scheme.

- **Put result in variable**: here you can define a variable to store the results. In case multiple files are uploaded, this variable is an array of the results, one entry for each file. If the returned JSON cannot be stored in a variable, an error will be generated.
- **Put response headers in variable**: here you can define an optional variable into which the HTTP response headers are stored. These headers are stored in an object where the key string maps to values which are arrays of strings. In case of multiple file upload, this variable is an array of the individual header objects, with one entry for each file.

# Call SOAP

With this node you can call a function on a SOAP server.

It has the following configuration options:

- URL: here you can define the URL of the SOAP server.
- WSDL: here you can define the location of the WSDL (Web Services Description Language).
- Function: here you can define the name of the function to call.
- Variable name: here you can define the parameters that are provided to the function. This has to be an object. Select the pencil to open Expression Builder.
- **Put result in variable**: here you can define a variable to store the results. In case multiple files are uploaded, this variable is an array of the results, one entry for each file.
- Convert result to JSON: if you select this checkbox, the result will be in JSON format.

#### Download to file

With this node you can download data from a (remote) server and save it to a file.

It has the following configuration options:

- **Download URL**: here you can specify the data to download. The supported protocol is HTPP. Select the pencil to open Expression Builder.
- Save As: here you can specify the name of the file where the data should be saved into. Select the pencil to open Expression Builder.

#### Upload file

With this node you can upload the contents of a file to a (remote) server.

It has the following configuration options:

- File to upload: here you can specify the file of which the data should be uploaded. Select the pencil to open Expression Builder.
- Upload URL: here you can specifies the URL where the data should be uploaded. FTP is the supported protocol.
- Username: here you can specify the user name if authentication is necessary.
- **Password**: here you can specify the password if authentication is necessary.

# **External Tools**

### Process By HotFolder Application

With this node you can save one or more files in a hotfolder of an external application, and wait for the results to return.

- File to process: here you can select the file that you want to process. Select the pencil to open Expression Builder.
- **Output File**: here you can specify the name that should be assigned to the output file. Options:
  - Leave this field blank: in this case the file name(s) as generated by the HotApp are used. This means that if the HotApp generates one file with the same name as the input file, the input file will be overwritten/updated

with the results of the HotApp. If the HotApp generates files with different name(s) (for example with a different extension), the output file(s) will be copied next to the input file, which remains unchanged.

- **Only specify a file name**: in case you specify the original file name, the input file will be overwritten, regardless of the naming convention used by the HotApp. If you specify a different file name, the output file will be copied next to the input file.
  - =

**Note:** You cannot use this setup when the HotApp generates more than one output file, since all files will be copied next to the input file with the same name, overwriting each other.

- **Only specify a folder**: in this case all the output file(s) will be copied into this folder, using the naming convention used by the HotApp.
- Specify a folder and file name: in this case all the output file(s) will be copied to the specified file.

**Note:** You can not use this setup when the HotApp generates more than one output file, because all files will be copied to the same location, overwriting each other.

• **Ignore Results**: if you select this checkbox, the results are ignored. You can choose this option if you are not interested in the actual results but you want to know that the process has ended correctly. This is for example useful to wait on a hard copy device. If you leave this checkbox unselected, the results generated by the application will be used.

# Input and Output folders of external application

- Mode: here you can specify how the application generates result files. Options:
  - In To Success: in this case the output is always written in the same folder.
  - In To Success or Error: in this case the output is written in a success folder or in an error folder.
  - In To Success, Warning or Error: in this case the output is written in a success folder, a warning folder or in an error folder.

The result file will be searched in one of the configured output folders. The location where the output is found will determine the path of the workable.

- **Input Hot Folder**: here you can specify the location where the input file should be copied to so that the hot folder application can find it for processing.
- **Insert (unique) folder**: if you select this checkbox, the input file is saved inside a unique subfolder in the hot folder input, so that the output can be searched in a folder with the same name (allowing for alternative output names and/or multiple output files).
- Success Output Folder: here you can specify the location where the hot folder application saves the file when processing was successful.

# Match results from external application

- Name Match Mode: here you can specify how the generated files should be found. Options:
  - File, Exact Match: if you select this option, the generated file name will exactly match the specified value.
  - Enclosing Folder, Exact Match: if you select this option, the enclosing folder of the generated file will exactly match the specified value.
    - **Note:** This option/value will be ignored when checkbox **Insert Folder** is selected.
  - File, Starts With: if you select this option, the generated file name will exactly start with the specified value.
  - File, Ends With: if you select this option, the generated file name will exactly end with the specified value.
  - File, Contains: if you select this option, the generated file name will contain the specified value.
  - File, Matches Regular Expression: if you select this option, the generated file name will match a regular expression with the specified value.

**Note:** You can use https://regex101.com as an online regex tester.

**Note:** For all modes, there is also an as **URL** version. In this case, the contents of the field **Name Match String** should be in URL notation. This is useful when you construct the name to match from the incoming file references which are typically in URL notation.

- Name Match String: here you can specify the name matching string. Options:
  - Leave this field blank in combination with all Name Match Modes excluding Folder Exact Match: if you select this option, the system will search for a (single) output file that contains a part that matches exactly with the name of the input file.
  - A Value in combination with all Name Match Modes excluding Folder Exact Match: if you select this option, the system will search for a (single) output file that contains a part that matches exactly with the specified value.
  - A Value in combination with all Folder Exact Match: if you select this option, the system will search for one or more output file in an enclosing folder that has a name that matches exactly with the specified value. :

=

Note: This option/value will be ignored when Insert Folder is selected.

Select an as URL in the field Name Match Mode if the supplied string is URL encoded.

- **Polling Interval**: here you can specify how often the system should check for generated results.
- Settling Time: here you can specify how long the system should ignore any changes in the generated results before we actually use the results. No changes can mean that the file modification date does not change, and/or no new files are found.
- **Time Out**: here you can specify how long the system should wait for results before marking this workable as failed.

# **Run Command Line Tool**

With this node you can run a command line tool.

It has the following configuration options:

- **Command**: here you can specify the command to run. Select the pencil to open Expression Builder. There are two typical use cases:
  - Run a system command that can be found in the path of the system. For example **ls** on OS X or **dir** on Windows. Specify the command to run.
  - Run an application of which you know the location on the file system. Specify the CLOUDFLOW path to the executable to run.
- Arguments: here you can specify the arguments to pass to the command line tool. Select + to add an argument and to delete an argument. Select the pencil to open Expression Builder.
  - **Note:** By default all references to files will be passed as CLOUDFLOW paths. You need to specify that they have to be converted to local file paths if you want get the references in the notation of the OS where the tool runs. You need to do that for each argument that needs to be converted since the system does not know which parameters are paths.
- Tool Generates Output Files: If you select this checkbox, the command line tool will generate output files.
- Save Console output: here you can specify if the command generates output on the console that you want to save for later use in the flow. If you select this checkbox, the output will be saved.

**Note:** You cannot select both the **Tool Generates Output Files** and the **Save Console output** checkboxes at the same time.

• Save Console error output: here you can specify if the command generates error output on the console that you want to save for later use in the flow. If you select this checkbox, the error output will be saved.

**Note:** You cannot select both the **Tool Generates Output Files** and the **Save Console error output** checkboxes at the same time.

- Save Console Output As: here you can define how the console (error) output will be saved. Options:
  - **JSON**: if you select this option, the output is saved as an array of strings where each string contains one line of the output.
  - Text: if you select this option, the output is saved as a single string containing all the lines.
  - File: if you select this option, the output will be streamed to the specified file.
- File for Console Output: here you can specify where the output should be saved on disk.

- File for Error Console Output: here you can specify where the output should be saved on disk.
  - Note: You can set the fields File for Console Output and File for Error Console Output to the same value, which means that both outputs will be saved to the same file. However, the order of the lines is not predictable.
- Append new output data to existing files: if you select this checkbox, the output and/or error-output will be appended to an already existing file. If you leave this checkbox unselected any existing files are cleared before running the command line tool.
- Working Directory: here you can specify the working directory to use when invoking the command line tool. You can leave this field empty if you do not want to specify the working directory.
- Max Concurrent Workers: here you can specify the number of concurrent calls that can be executed to the command line tool. This can be useful if you do not want to lock your system on this node if the called command line tool will be running for a long time.

If you set the value to **0**, the concurrency is not limited. If you set it to **1**, this node will only be executed once at the same time. If you set it to **2**, this node will be executed maximum twice at the same time, etcetera.

# **Output device**

### Print

With this node you can print one or more files.

- File to process: here you can specify the files to print.
- **Printer Name**: here you can specify the printer.
- **Paper ID**: here you can specify the paper to use. If you leave this field empty, the smallest paper for the selected printer that can be used to print the file will be automatically selected.
- Orientation: here you can specify the orientation when positioning the file on the paper. Options:
  - **Portrait**: if you select this option, the file will be placed on the paper without rotation. This is the default value.
  - Landscape: if you select this option, the file will be placed on the paper with a rotation of 90 degrees.
  - Auto: if you select this option, the file will be placed on the paper with the orientation that is optimal to place the file.
- **Duplex Printing**: here you can specify if (and how) a multi-page file should be printed on the front-and-back of the paper when printing. Options:
  - Simplex: if you select this option, only print the front of the paper will be printed.
  - **Duplex**: if you select this option, the front and back of the paper will be printed. The page will not be tumbled on the back, which means that both sides have the same orientation, as in most books.
  - **DuplexTumble**: if you select this option, the front and back of the paper will be printed. The page will be tumbled on the back, which means that the pages are flipped vertically, as in a notepad.
- Page box to print: here you can select the page box to print.
- Scale Mode: here you can specify how the file must be scaled when printing. Options:
  - Fixed: if you select this option, the file will be scaled as defined by the scale option. This is the default value.
  - Scale To Fit: if you select this option, the file will be scaled to print as large as possible on the paper.
  - Scale To Fit Shrink Only: if you select this option, no scaling will be applied if the file fits on the paper as is. If not, the file will be scaled to print as large as possible on the paper.
- Scale: here you can specify the (fixed) scaling to use while printing the file. 1 means no scaling.
  - **Note:** This value will only be used when **Scale Mode** is set to **Fixed**.

- **Apply distortion**: here you can specify how distortion (and scaling) information in the file must be handled when printing. Options
  - Always: if you select this option, the file will be printed with distortion applied.
  - Never: if you select this option, the file will be printed without distortion.
  - As Is: if you select this option, the file will be printed as is.:
    - If the file is distorted, it will be printed with this distortion.
    - If the file only contains distortion instructions, it will be printed without distortion.
- Only Use PrintableArea: if you select this checkbox, only the content inside the printable area is used to print the file. You can leave this checkbox unselected if it is not a problem when your document contains information that would be printed in the margin area where the printer cannot put any ink on the paper. This option is only used when searching papers, that is when **Paper ID** is empty. By default, the complete document will be positioned inside the printable area of the printer-paper to print the complete job.
- Margin: here you can specify the a margin on the paper that will not be used for printing when determining the orientation and scaling. This parameter will be ignored when set to **0** or when set to a negative value.
- **Resolution**: here you can specify the resolution that will be used to rasterize PDF files and other non-raster files. The rasterized data will be sent to the printer. The generated raster data is 8 bits per pixel RGB. Best practices:
  - Select a resolution that preserves the needed detail but that is not too high.
  - Select a resolution that aligns properly with the resolution of the printer (for example, use 300 DPI on a 1200 DPI printer).

# Conversions

# Create HelioDisk JobTicket

With this node you can create a Hell-Gravure HelioDisk production (.pro) file.

It has the following configuration options:

- **TIFF Files**: here you can specify the TIFF files that you want to engrave. Select the pencil to open Expression Builder.
- HelioDisk .PRO File: here you can specify the output path and name of the generated production file. If you leave this field blank, an output file with the original name with extension .pro will be written in the same folder as the input TIFF file. The used file path follows the standard input-output file name combination rules in CLOUFLOW. See Output File Path Generation on page 468 for more information.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- **Convert to Job Ticket**: if you select this checkbox, the **HelioDisk** application is called to convert the production file into a HelioDisk Job Ticket.



**Note:** The logical HELIODISK\_PROGRAM needs to be configured correctly to use this option. You need to configure it to point to HelioDisk.

- Job Name: here you can specify the name to assign to the job (inside the production file). If you leave this parameter blank, the original name without extension will be used.
- Job Comment: here you can specify a comment to assign to the job.
- **Print Material**: here you can define the material the job will be printed on. This parameter is only used to find the predefined **Test Cut Data** inside HelioDisk.
- Set All Parameters: if you select this checkbox, all parameters will be set in the production file. That means that all parameters will be set when the file is opened in HelioDisk, even if they are (incorrect) default values. It is the operator's responsibility to find the parameters that still need to be configured. If you leave this checkbox unselected, parameters that are default and/or empty will not be set in the production file. That means that the operator still has to run over the unspecified options to configured them after opening the file in HelioDisk.

# **Defaults for all Cylinders**

- Engraving method: here you can select the default engraving method to use for all Cylinders.
- Engraving formation: here you can select the (default) engraving formation to use for all Cylinders.
- Mirror in circumferential direction: if you select this checkbox, the default mirroring for all cylinders will be set to circumferential direction.
- Screen Value: here you can select the (default) screening value to use for all cylinders.
- Screen Angle: here you can select the (default) screening angle to use for all cylinders. This parameter is ignored when the field Screen Angle-list is not empty.
- Screen Angle-list: here you can select the sequence of screening angle to use for cylinders. This is a comma separated list of angles for each subsequent cylinder. The first value in the list will be assigned to the first cylinder, the second value to the second cylinder, etc. The first (and next) values will be used again if you have more cylinders than entries in this list. For example, if your list contains 1,2,3 and you have 4 cylinders, 1 will also be assigned to the fourth cylinder.
- Gradation file: here you can select the (default) gradation file to use for all cylinders.
- **GBS offset**: here you can select the (default) GBS offset to use for all cylinders.
- Cylinder Name: here you can select the (default) cylinder name to use for all cylinders.
- Cylinder Length: here you can select the (default) cylinder length to use for all cylinders.
- Quality, Sharpen Axial: here you can select the (default) quality sharpen value in axial direction to use for all cylinders.
- Quality, Sharpen Circumference: here you can select the (default) quality sharpen value in circumference to use for all cylinders.
- Quality, Font Width (interpolation file): here you can select the (default) interpolation file for font width to use for all Cylinders.
- **Gradations folder**: here you can select a (default) folder where gradation files can be found. If you (always) configure the gradation files with a file URL, you can leave this setting empty. If you only specify a name for the gradation files, you can specify this location.
- Font Width folder: here you can select a (default) folder where interpolation files for font width can be found. If you (always) configure the interpolation files with a file URL, you can leave this setting empty. If you only specify a name for the interpolation files, you can specify this location.

# Job Details

- Use Job Details: if you select this checkbox, the detailed job and cylinder info will be read from specified data. This data is a JSON file that complies with a fixed data structure. See Data Structure of HelioDisk Job Details on page 462 for more information on the data structure of the HelioDisk Job Details.
- Job Details: here you can select the detailed job and cylinder info.
- Sort and match Job Details and plates: with this drop-down list you can select the order of the cylinders in the .pro file. Options:
  - Sort by Job Details: if you select this option, the cylinders will be sorted according to the cylinder order in the job details. The information derived from the plates will be re-ordered to match that order (based on the separation names).
  - **Sort By Plates**: if you select this option, the cylinders will be sorted according to the plate order. The information found in the job details will be re-ordered to match that order (based on the separation names).
  - Sort By Index: if you select this option, the cylinders will be sorted *as is*. In this case the amount of cylinder entries in the job details is expected to be the same as the amount of plates.

# **Text Cut Layout**

- X Origin: here you can specify the x origin of the test cut (see field X origin base).
- Y Origin base: here you can specify if the origin of the test cut layout is relative to the cylinder or to the engraving.
- Y Origin: here you can specify the y origin of the test cut (see field Y origin base).
- X Origin base: here you can specify if the origin of the test cut layout is relative to the gbs or to the picture.

- Width: here you can specify the width of the test cut.
- **Height**: here you can specify the height of the test cut.
- Line Order: here you can specify the line order in the test cut.
- Single test cut width: here you can specify the width of a single cell in the test cut.
- Single test cut height: here you can specify the height of a single cell in the test cut.
- Line Length: here you can specify the line length to use in the test cut.

### **CSV to JSON**

With this node you can interpret CSV data and convert it to a JSON object.

- CSV File: here you can define the URL of the CSV file that contains the data Select the pencil to open Expression Builder.
- Conversion Mode: here you can specify how to read or convert the CSV data. Options:
  - **Row By Row**: if you select this option, the data is read **row by row** and the results are stored as **records**. Each record contains an object where you can access the data **by key** (derived from the data in the first row).
  - Column By Column: if you select this option, the data is read column by column and the results are stored as records. Each record contains an object where you can access the data by key (derived from the data in the first column).
  - As Grid: if you select this option, the data is read and the results are stored as three nested arrays/objects, where the first level are the sheets, the second level are the rows, and the third level are the columns. The sheet level will not be present if you explicitly select a sheet.
- Value Separator: here you can define the character that is used to separate the different values.
- Field Names: here you can specify the expected row or column names in the first record. Only columns or rows that match the specified names will be extracted.
  - If no field names are specified, all non-empty rows or columns will be read.
  - When the Conversion Mode is As Grid, this parameter is ignored.
- No Field Names:
  - If the CSV data has no row or column names in the first line, you need to select this checkbox. In this case, you need to specify the Field Names. Those names will be used to create the JSON data. If the CSV data has column names, you need to deselect this checkbox. When the Conversion Mode is **As Grid**, this parameter is ignored.
- Skip First: if you want to exclude the first line from the data saved to JSON, you need to select this checkbox, even though the data in the first row will still be used to check/define the field names. If you want to include the first line, you need to deselect this checkbox. When the Conversion Mode is As Grid or when the checkbox No Field Names is unselected, his parameter is ignored.
- JSON Output: here you can specify the output path of the JSON data.
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an absolute CLOUDFLOW URL containing a filename (cloudflow: //..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you define an absolute CLOUDFLOW URL without a filename (cloudflow: //..., see Server URL for more information), an output file with the original file name will be written in the specified folder.
  - If you enter a **relative path containing a file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
  - If you enter a **relative path without a file name**, an output file with the original file name will be written in the specified folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# **Excel to JSON**

With this node you can interpret Excel data and convert it to a JSON object.

It has the following configuration options:

• Excel Data: here you can define the URL of the excel file that contains the data. Select the pencil to open Expression Builder.

**Note:** Only Excel files with extension **.xls** are supported.

- Conversion Mode: here you can specify how to read or convert the Excel data. Options:
  - **Row By Row**: if you select this option, the data is read **row by row** and the results are stored as **records**. Each record contains an object where you can access the data **by key** (derived from the data in the first row).
  - Column By Column: if you select this option, the data is read column by column and the results are stored as records. Each record contains an object where you can access the data by key (derived from the data in the first column).
  - As Grid: if you select this option, the data is read and the results are stored as three nested arrays/objects, where the first level are the sheets, the second level are the rows, and the third level are the columns. The sheet level will not be present if you explicitly select a sheet.
- Sheet Name: here you can define the sheet name to get the data from.
  - If you set this parameter to an empty string and if the Conversion Mode is **Row By Row** or **Column By Column**, the first sheet will be used.
  - If you set this parameter to an empty string and if the Conversion Mode is As Grid, all sheets will be returned.
- Field Names: here you can specify the expected row or column names in the first record. Only columns or rows that match the specified names will be extracted.
  - If no field names are specified, all non-empty rows or columns will be read.
  - When the Conversion Mode is As Grid, this parameter is ignored.
- No Field Names:
  - If the Excel data has no row or column names in the first line, you need to select this checkbox. In this case, you need to specify the Field Names. Those names will be used to create the JSON data. If the Excel data has column names, you need to deselect this checkbox. When the Conversion Mode is **As Grid**, this parameter is ignored.
- Skip First: if you want to exclude the first line from the data saved to JSON, you need to select this checkbox, even though the data in the first row will still be used to check/define the field names. If you want to include the first line, you need to deselect this checkbox. When the Conversion Mode is As Grid or when the checkbox No Field Names is unselected, his parameter is ignored.
- **Support Unicode**: if you select this checkbox, Unicode text (text with for example accented letters) is supported. This option will however prevent some complex files to be processed.
- JSON Output: here you can specify the output path of the JSON data.
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL containing a filename** (cloudflow://..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you define an absolute CLOUDFLOW URL without a filename (cloudflow: //..., see Server URL for more information), an output file with the original file name will be written in the specified folder.
  - If you enter a **relative path containing a file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
  - If you enter a **relative path without a file name**, an output file with the original file name will be written in the specified folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### Image to PDF

With this node you can convert an image file to a PDF file.

It has the following configuration options:

- **Image File**: here you can select the image file you want to convert. This can be a single file or a set of files (for example TIFF files from a RIP output) that will be combined into a single PDF file.
- Stacked Files: here you can specify how to convert the image file.
  - If you don't select this checkbox, the (single) file will be converted **as is** into a PDF file. This means that a black and white or grayscale image (for example a TIFF file) will be converted to a black and white or grayscale image in the final PDF.
  - If you select this checkbox, multiple image files are combined into a single PDF file. This also changes the output for a single black and white or grayscale image, because it will be output as a single channel colored with any metadata that can be found in the image file.
- **PDF File**: here you can specify the name to assign to the PDF file. If you leave this field blank, the original file name but with extension **pdf** in the same folder as the input image file(s) will be used. The used file path follows the standard input-output file name combination rules in CLOUFLOW. See Output File Path Generation on page 468 for more information.
- Lossless compression: The image data in the generated PDF file is normally compressed with JPEG, and hence contains compression artifacts. This might be a problem if the saved file (only) needs to be printed afterward. Enable this option to remove any compression artifacts (though compression artifacts in the initial data might still be visible in the output)
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Len To TIFF

With this node you can convert a LEN file to a TIFF file.

It has the following configuration options:

- Len File: here you can specify the LEN files you want to convert. Select the pencil to open Expression Builder.
- **TIFF File**: here you can specify the output path and name of the TIFF file. If you leave this field blank, an output file with the original name with extension .tiff will be written in the same folder as the input LEN file. The used file path follows the standard input-output file name combination rules in CLOUFLOW. See Output File Path Generation on page 468 for more information.
- **TIFF compression**: here you can select the compression for the data in the TIFF file.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Save Preview To File

With this node you can save a preview of a file.

It has the following configuration options:

- Input File: here you can specify the files where you want to generate a preview from. Select the pencil to open Expression Builder.
- **Page**: here you can select the page from the file. If you set the value to **0**, the preview is generated from the first page.
  - **Note:** You need to set the value to **0** as well for single page documents.
- **Output File**: here you can specify the result names for the generate preview file(s). If you leave this field **blank**, a file with a default name (this is, the original name but with the correct extension for the selected image format) will be written in the same folder as the input file. The used file path follows the standard input-output file name combination rules in CLOUFLOW. See Output File Path Generation on page 468 for more information.

You can either specify one or several values in this field.

- In case you specify one value, this value will be combined with each input file to generate an output file.
- In case you specify several values, each value will be combined with each input file to generate an output file.

- Mode: here you can specify the data to use when generating the preview data. Options:
  - **Data From Thumbnail**: in this case the saved preview will be based on the thumbnail calculated by CLOUDFLOW. This thumbnail is always 512 pixels and you cannot change it. This is a fast operation since the preview is obtained from the database.
    - **Note:** Using this option with PNG will result in files with compression artifacts because the thumbnail is stored in the database as a JPEG file.
  - **Calculate Preview**: in this case the saved preview will be calculated, and you can select the size. This can be much slower than **Data From Thumbnail** because the node needs to render the file.
- Image Type: here you can specify the image format to use when saving the preview (JPEG of PNG).
- **Preview Size**: here you can specify the size of the preview. This option is not available when **Mode** is set to **Data From Thumbnail**.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

#### Save Rasterized To File

With this node you can rasterize a file and save it to a file.

It has the following configuration options:

- Input File: here you can specify the files you want to rasterize. Select the pencil to open Expression Builder.
- **Output File**: here you can specify the output path and name of the rasterized file. If you leave this field blank, an output file with the original name with the correct extension will be written in the same folder as the input file. The used file path follows the standard input-output file name combination rules in CLOUFLOW. See Output File Path Generation on page 468 for more information.
- **Resolution**: here you can specify the resolution that will be used to rasterize the data. The generated raster data is 8 bits per pixel RGB.
- Render Anti Aliassed: if you select this checkbox, the file will be rendered with anti-aliasing.
- Lossless compression: if you do not select this checkbox, the image data in the generated PDF file is compressed using JPEG, and may contain compression artifacts. This might be a problem if the saved file needs to be printed. If you select this checkbox, any compression artifact is removed.

**Note:** Compression artifacts in the initial data might still be visible in the output.

• **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# XML to JSON

With this node you can interpret XML data and convert it to a JSON object.

- XML Data Type: here you can define how the contents of the XML Data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the content of the XML data parameter will be detected automatically.
  - XML data (from other node): if you select this option, the content of the XML data parameter will be used as XML data.
  - XML string: if you select this option, the content of the XML data parameter will be used as an XML string.
  - URL reference: if you select this option, the content of the XML data parameter will be used as an URL reference.
- XML Data: here you can specify the XML data. Select the pencil to open Expression Builder. Options:
  - XML data generated by a previous node.
  - XML in a string.
  - an URL reference to a XML file.
  - **Note:** See also XML Data Type.

- XML Template Type: here you can define how to interpret the XML template field. Options:
  - Auto detect: if you select this option, the data are interpreted as an asset in case of a CLOUDFLOW or file URL. If it is not a URL, the data is interpreted as XML data.
  - XML Data: if you select this option, the data is always interpreted as XML data.
  - Cloudflow or file URL: if you select this option, the data is always interpreted as a URL.
  - No Template: if you select this option, the XML is interpreted without using a template.
- XML Template: here you can enter the XML template; or a CLOUDFLOW or file URL to an asset containing the template.
- Allow Incomplete Template: if you leave this checkbox unselected, the XML to JSON node will only process XML data that fully matches the template. If you select this checkbox, the node will process XML data that contains data which is not defined in the template. In this case, undefined data is ignored.
- Strip White space: if you select this checkbox, white spaces are stripped.
- Store result in variable: if you select this checkbox, the resulting JSON data is stored in a variable. If you do not select it, the data is stored in a data container.
- **Result variable name**: here you can define the variable into which the results are put in case you have selected the checkbox **Store result in variable**.
- **Output**: here you can specify the output path of the JSON data in case you have not selected the checkbox **Store** result in variable. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an absolute CLOUDFLOW URL containing a filename (cloudflow: //..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you define an absolute CLOUDFLOW URL without a filename (cloudflow: //..., see Server URL for more information), an output file with the original file name will be written in the specified folder.
  - If you enter a **relative path containing a file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
  - If you enter a **relative path without a file name**, an output file with the original file name will be written in the specified folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

Note: Update

Be sure to update this node and its XML template regularly.

# Share

=

# Add Sync Spec Mapping

With this node you can add a mapping to a sync spec.

It has the following configuration options:

- Sync spec: here you can define the sync spec to modify.
- Name: here you can specify a name to identify the mapping.
- Source site: here you can define the site from which the files are synchronized.
- Source folder: here you can specify the CLOUDFLOW URL (cloudflow: //..., see Server URL for more information) of the folder to synchronize. This folder is located on the source site.
- Target folder: here you can specify the CLOUDFLOW URL (cloudflow: //..., see Server URL for more information) of the folder where the synchronized files will be put. This folder is located on the target site, which is specified in the sync spec.
- Filter: here you can specify a Regular expressions on page 475 indicating which files need to be synchronize. If you leave this field empty, all files in the source folder will be synchronized.

**Note:** You can use https://regex101.com as an online regex tester.

- Direction: here you can define the direction in which the files are synchronized. Options:
  - Bidirectional: if you select this option, changes from both source and target folders are synchronized.
  - Source → Target: if you select this option, only changes from the source site are synchronized. Changes at the target site are discarded.
  - Target → Source: if you select this option, only changes from the target site are synchronized. Changes at the source site are discarded.
  - **Disabled**: if you select this option, nothing is synchronized.
  - **Propagate file deletions**: if you select this checkbox, file deletions will also be propagated to the other site. This checkbox is only available if you selected **Source**  $\rightarrow$  **Target** or **Target**  $\rightarrow$  **Source**.

# Remove Sync Spec Mapping

With this node you can remove a mapping from a sync spec and clean up the target folder.

It has the following configuration options:

- Sync spec: here you can specify the sync spec to modify.
- Mapping: here you can specify the name of the mapping.

### Sync File

With this node you can synchronize files between sites.

It has the following configuration options:

- File to synchronize: here you can specify the file to synchronize. Select the pencil to open Expression Builder.
- Source site: here you can specify the source site.
- Target cloudflow URL: here you can specify the target cloudflow URL.
- Target site: here you can specify the target site.
- **Retry on server error**: if you select this checkbox, the system automatically retries to synchronize in case of a server error.
- **Remove similar workables**: if you select this checkbox, redundant *workable* on the same whitepaper that send the same source file to the same destination are canceled.

#### Sync Folder

With this node you can synchronize a folder from one SHARE site to another.

Select the pencil to open Expression Builder.

- Folder to synchronize: here you can specify the folder you want to synchronize. Select the pencil to open Expression Builder.
- Source site: here you can define the source site of the folder.
- Target cloudflow URL: here you can define the CLOUDFLOW URL of the target folder.
- Target site: here you can define the target site.
- **Only copy contents**: if you select this checkbox, only the content in the **Folder to synchronize** is synchronized. If you leave this checkbox unselected, both the content in **Folder to synchronize** and the folder itself are synchronized.
- **Remove files**: if you select this checkbox, files and folders in the target folder that are missing in the source folder are also removed.
- Simultaneous transfers: here you can specify the number of files that you can transfer at the same time.
- Settle Time: here you can select the settle time. If the value is not equal to 0, the workflow will only continue if nothing has changed in the Folder to synchronize during that period of time after the last synchronization.
- **Run asynchronously**: if you select this checkbox, the node runs asynchronously from the sync call to the initiating site. This is to prevent issues with network response time-outs.

# Update Sync Spec Mapping

With this node you can remove a mapping from a sync spec and clean up the target folder.

- Sync spec: here you can specify the sync spec you want to update.
- Mapping: here you can specify the name of the mapping.
- Direction: here you can define the sync mode. Options:
  - **Bidirectional:** if you select this option, changes from both source and target folders are synchronized.
    - Source → Target: if you select this option, only changes from the source site are synchronized. Changes at the target site are discarded.
    - Target → Source: if you select this option, only changes from the target site are synchronized. Changes at the source site are discarded.
    - **Disabled**: if you select this option, nothing is synchronized.
    - **Propagate file deletions**: if you select this checkbox, file deletions will also be propagated to the other site. This checkbox is only available if you selected **Source** → **Target** or **Target** → **Source**.
- **Propagate file deletions**: if you select this checkbox, file deletions will also be propagated to the other site. This checkbox is only available if you selected **Source** → **Target** or **Target** → **Source**.

#### Wait for Sync

With this node you can synchronize a file between sites.

It has the following configuration options:

- Sync Spec: here you can specify the sync spec.
- **Mapping**: here you can define the mapping (optional).
- Settle Time: here you can define the minimum time that should elapse before the actual processing begins.

### Data

Data nodes create and/or use data.

CLOUDFLOW uses the concept of data. Contrary to files,

- data are not saved on a disk, but are stored in a container in the database. A container is a location in the database where nodes store their resulting (temporary) data.
- data are never overwritten, which means you can always use the created data in any further step of the workflow.

#### Example

In the workflow underneath, the **Create XML** node is used. In general, this node creates XML data or file(s), depending on the content of the **Output** field:

- If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
- If you define an absolute CLOUDFLOW URL (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.

In the example, the output field is empty. Consequently, no physical XML file is created and the XML data are stored in a container in the database. These data are used in the consequent node (XML to JSON) as the input data.



### List SQL Records

With this node you can list records from a database by running an SQL query.

It has the following configuration options:

- SQL Query: here you can specify the SQL query to execute to obtain the list of records you want.
- Database URL: here you can specify the JDBC URL where the database can be reached.
- User Name: here you can specify the user name to use while logging in to the database.
- **Password**: here you can specify the password to use while logging in to the database.
- JSON Output: here you can specify where to save the JSON data. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- **Note:** Make sure to add/enable the **QUANTUM JAVA** Worker to see this node.

# **Call Custom Function On Data Connector**

With this node you can call a custom function on a data connector.

- **Data Connector**: here you can specify the base URL where the data connector is located in the network. There are two options:
  - If the data connecter is hosted internally in CLOUDFLOW, it should contain the name of the data connector.
  - If the data connector is hosted externally, it should include:
    - The address of the workstation the data connector is on.
    - The base path to the data connector (for example http://mybox.mydomain.com:8080/ MyDataConnector).
- Function: here you can specify the Function you want to call on the Data Connector (for example /MyTable/ MyFunction)

- **Parameters**: here you can specify the parameters you want to provide to the function. Select + to add a function. Select the pencil to open Expression Builder.
- JSON Output: here you can specify where to save the JSON data. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Create XML

Wit this node you can create XML data or an XML file.

It has the following configuration options:

- **Output**: here you can specify the output path of the XML file. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.
- XML Data: here you can enter the XML content. Select the pencil to open Expression Builder.

# Get Record From Data Connector

With this node you can get a record from a database through a data connector.

- **Data Connector**: here you can specify the base URL where the data connector is located in the network. There are two options:
  - If the data connecter is hosted internally in CLOUDFLOW, it should contain the name of the data connector.
  - If the data connector is hosted externally, it should include:
    - The address of the workstation the data connector is on.
    - The base path to the data connector (for example http://mybox.mydomain.com:8080/ MyDataConnector).
- **Table**: here you can specify the table name you want to use in the data connector, which maps to a table in the database.
  - **Note:** This is the assigned name in **CloudBuilder**, which does not always map exact to the table name in the database. In most cases the only difference is the capitalization.
- Record ID: here you can specify the id of the record you want to get from the data connector.
- JSON Output: here you can specify where to save the JSON data. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# List Records From Data Connector

With this node you can list records from a data connector.

It has the following configuration options:

- **Data Connector**: here you can specify the base URL where the data connector is located in the network. There are two options:
  - If the data connecter is hosted internally in CLOUDFLOW, it should contain the name of the data connector.
  - If the data connector is hosted externally, it should include:
    - The address of the workstation the data connector is on.
    - The base path to the data connector (for example http://mybox.mydomain.com:8080/ MyDataConnector).
- **Table**: here you can specify the table name you want to use in the data connector, which maps to a table in the database.
  - **Note:** This is the assigned name in **CloudBuilder**, which does not always map exact to the table name in the database. In most cases the only difference is the capitalization.
- **Query**: here you can specify the query you want to run in the data connector to get the records you want. Select + to add a query. Select the pencil to open Expression Builder.
- Order By: here you can specify the fields to order by, and whether they should be sorted ascending or descending. Select + to add a field. Select the pencil to open Expression Builder.
- Fields: here you can specify the fields to get. Select + to add a field. Select the pencil to open Expression Builder.
- Limits
  - **First Record**: here you can specify the first object that matches the query that should be returned. This value is zero based, which means that 0 will return the first match as first result.
  - Maximum number of records: here you can specify the maximum records that should be returned.
- JSON Output: here you can specify where to save the JSON data. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Transform XML

With this node you can transform an XML file using XSLT or XQuery into another XML or JSON file.

It has the following configuration options:

- XML Data: here you can specify the XML data. Select the pencil to open Expression Builder. Options:
  - XML data generated by a previous node.
  - XML in a string.
  - An URL reference to a XML file.

# **Note:** See also XML Data Type.

- XML Data Type: here you can specify how the contents of the XML data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the content of the XML data parameter will be detected automatically.
  - XML data: if you select this option, the content of the XML data parameter will be used as XML data.
  - XML string: if you select this option, the content of the XML data parameter will be used as an XML string.
  - URL reference: if you select this option, the content of the XML data parameter will be used as an URL reference.

- **Transformation file**: here you can define the transformation file. This can either be an XSLT file or a XQuery file. You need to the format in **Transformation Type**.
- **Transformation Type**: here you can specify the format of the transformation file. Options:
  - **XSLT 1.0**: xslt 1.0 with xpath 1.0
  - XSLT 2.0: xslt 2.0 with xpath 2.0
  - XSLT 3.0: xslt 3.0 with xpath 3.0
  - **XQuery 1.0**: xquery 1.0 with xpath 2.0
  - XQuery 3.0: xquery 3.0 with xpath 3.0
- **Output**: here you can specify where to save the transformed output. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you enter a **file name without an extension**, an output file with the specified file name and the original extension will be written in the same folder as the input file.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name and the original extension will be written in the specified folder.
  - If you enter a **relative path without a file name and extension**, an output file with the original file name and extension will be written in a folder based on the original input folder.
  - If you enter a **relative path containing a file name and extension**, an output file with the specified file name and the original extension will be written in a folder based on the original input folder.
- **Output Type**: here you can define the output type should be XML or JSON.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Update JSON

With this node you can create a JSON file or you can update a JSON file by updating nodes that are selected by a path.

It has the following configuration options:

- JSON Data: here you can specify the JSON data. Select the pencil to open Expression Builder.Options:
  - JSON data generated by a previous node.
  - JSON from a variable.
  - A URL reference to a XML file.

# **Note:** See also **JSON Data Type**.

- JSON Data Type: here you can specify how the contents of the JSON data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the data type will be detected automatically.
  - JSON Data (output from other node): if you select this option, the JSON content will be interpreted as JSON data.
  - JSON Data (for example from a variable): if you select this option, the JSON content will be interpreted as JSON data.
  - URL reference: if you select this option, the JSON content will be interpreted as an URL reference.
- Path: here you can define an array of elements to update. For each entry you need to define:
  - A **path**: here you can define a JSON path that selects an element in the JSON document. Select the pencil to open Expression Builder.
  - A value: here you can specify the value to set. Select the pencil to open Expression Builder.

Select + to add more paths.

- **Output**: here you can specify where to save the updated output. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you enter a **file name without an extension**, an output file with the specified file name and the original extension will be written in the same folder as the input file.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name and the original extension will be written in the specified folder.
  - If you enter a **relative path without a file name and extension**, an output file with the original file name and extension will be written in a folder based on the original input folder.
  - If you enter a **relative path containing a file name and extension**, an output file with the specified file name and the original extension will be written in a folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### Update Record To Data Connector

With this node you can update a record in a database through a Data Connector.

It has the following configuration options:

- **Record Data Type**: here you can specify how the contents of the JSON data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the code will examine the content of the provided data, and then use it either as JSON Data or as a URL.
  - **JSON in a container (output from another node)**: if you select this option, the content will be interpreted as content in a container, coming from another node.
  - JSON Data (e.g. from a variable): if you select this option, the content of the data will be interpreted as JSON Data.
  - URL Reference: if you select this option, the content of the data will be interpreted as a URL.
- **Record Data**: here you can specify the data you want to save in the data connector.
  - If you want to update a record, it **must** include the **id** field.
  - If you want to create a new record, **must not** include the **id** field.

Only the fields that you specify will be updated, all other fields in existing records will be unchanged. Options:

- JSON data generated by a previous node.
- A URL reference to a XML file.

# **Note:** See also **JSON Data Type**.

- **Data Connector**: here you can specify the base URL where the data connector is located in the network. There are two options:
  - If the data connecter is hosted internally in CLOUDFLOW, it should contain the name of the data connector.
  - If the data connector is hosted externally, it should include:
    - The address of the workstation the data connector is on.
    - The base path to the data connector (for example http://mybox.mydomain.com:8080/ MyDataConnector).
- **Table**: here you can specify the table name you want to use in the data connector, which maps to a table in the database.
  - **Note:** This is the assigned name in **CloudBuilder**, which does not always map exact to the table name in the database. In most cases the only difference is the capitalization.

- JSON Output: here you can specify where to save the JSON data. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### Update XML

With this node you can update an XML file by updating nodes that are selected by XPath.

It has the following configuration options:

- XML Data: here you can specify the XML data. Select the pencil to open Expression Builder. Options:
  - XML data generated by a previous node.
  - XML in a string.
  - An URL reference to a XML file.

### **Note:** See also XML Data Type.

- XML Data Type: here you can specify how the contents of the XML data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the content of the XML data parameter will be detected automatically.
  - XML data: if you select this option, the content of the XML data parameter will be used as XML data.
  - XML string: if you select this option, the content of the XML data parameter will be used as an XML string.
  - URL reference: if you select this option, the content of the XML data parameter will be used as an URL reference.
- **XPath**: here you can define the XPath expression that will select a specific node in the input XML.
- Value: here you can define new value for the selected node. Select + to add a path and value. Select the pencil to open Expression Builder.
- **Output**: here you can specify where to save the updated output. Options:
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you enter a **file name without an extension**, an output file with the specified file name and the original extension will be written in the same folder as the input file.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name and the original extension will be written in the specified folder.
  - If you enter a **relative path without a file name and extension**, an output file with the original file name and extension will be written in a folder based on the original input folder.
  - If you enter a **relative path containing a file name and extension**, an output file with the specified file name and the original extension will be written in a folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Get Custom Object Record

With this node you can get a record from a custom objects collection.

- Collection: here you can specify the name of collection you want to use. The name cannot include the part customobjects. For example: myCollection.
- Record ID: here you can specify the ID of the record you want to get.

- JSON Output: here you can specify where to save the JSON data.
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

### List Custom Object Record

With this node you can list records from a custom objects collection.

It has the following configuration options:

- Collection: here you can specify the name of collection you want to use. The name cannot include the part customobjects. For example: myCollection.
- **Query**: here you can specify the query you want to run on the collection to get the records you want. Select + to add a query. Select Select the pencil to open Expression Builder.
- Order By: here you can specify the fields to order on, and whether they should be sorted ascending or descending. Select + to add a field to order on. Select Select the pencil to open Expression Builder.
- Fields: here you can specify the fields to get. Select + to add a field. Select Select the pencil to open Expression Builder.
- Limits: First Record: here you can specify the first object that matches the query that should be returned. This value is zero based, which means that 0 will return the first match as first result.
- Maximum number of records: here you can specify the maximum number of records to return. If you set this value to 0, all matches will be returned.
- JSON Output: here you can specify where to save the JSON data.
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# **Update Custom Object Record**

With this node you can update a record in a custom objects collection.

- **Record Data Type**: here you can specify how the contents of the JSON Data parameter should be interpreted. Options:
  - Auto detect: if you select this option, the code will examine the content of the provided data, and then use it either as JSON Data or as a URL.
  - **JSON in a container (output from another node)**: if you select this option, the content will be interpreted as content in a container, coming from another node.
  - JSON Data (e.g. from a variable): if you select this option, the content of the data will be interpreted as JSON Data.
  - URL Reference: if you select this option, the content of the data will be interpreted as a URL.

- Mode: here you can decide if a record should be updated and/or created. Options:
  - Update: if you select this option, the specified data should update an existing record. This means that the data must contain a valid \_id field.
  - **Create**: if you select this option, the specified data should create a new record. This means that the data should not contain a valid **id** field.
  - Create or update: if you select this option, the specified data can create a new record or update an existing record. This depends on whether a valid id field is present or not.
- **Record Data**: here you can specify the data you want to save in the collection. Depending on the value of the **Mode** parameter, a valid **\_id** should or should not be present. The complete content of an existing record will be replaced by the specified data. Depending on the value of the **Record Data Type** parameter, the JSON can be:
  - A JSON data generated by a previous node.
  - A URL reference to a JSON file.
- **Collection**: here you can specify the name of collection you want to use. Make sure the name includes the database name (for example **nucleus.files**).
- JSON Output: here you can specify where to save the JSON data.
  - If you leave this field **blank**, the data will be sent to a container in the database and no physical file will be created.
  - If you define an **absolute CLOUDFLOW URL** (cloudflow: //..., see Server URL for more information), containing a file name and an extension, a physical output file with the specified file name will be created.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

# Jobs

With Jobs nodes you can build a Job workflow that drives Jobs templates. See JOBS on page 372 for more information on Jobs.

# Copy Job

With this node you can copy a Job.

- **Copy Job ID**: here you can define the job to copy. The new job will be a copy of the specified Job record in the database, keeping all predefined information, tasks etc.
- **Parent Job ID**: here you can enter a unique parent Job ID where the Job will be nested into. If you leave this field blank, the Job will stand on its own.
- Job Type: here you can select the type of Job. You can use this to store several different sets of Jobs in one database collection. You can leave this blank if your template Job already contains a type.
- **Identifier**: here you can define a (unique) identifier for this Job. This is not an auto assigned value, and there is no constraint on the contents. This value is intended to either provide a back-link to the Job in the master database or to provide a human way to reference the Job in an unique way. The fact that this key is unique is not enforced by CLOUDFLOW. The default value is an empty string.
- Name: here you can define a name of the Job.
- State: here you can define the (initial) state of the Job. You can leave this blank if your template Job already contains a type.
- **Description**: here you can define a description for the Job.
- Team name: here you can assign a team to the Job.

- Clear Files: if you select this checkbox, all file references from the copied Job will be removed. This does not have an impact on files on disk.
- Files: here you can selects (an initial) set of files for the Job.
- Files Tag: here you can define a tag to assign to the added files. You can leave this blank if you do not want to assign a tag to the files (that is, if all the files in your Job have the same meaning).
- Clear Folders: if you select this checkbox, all folder references from the copied Job will be removed. This does not have an impact on folders on disk.
- Folder: here you can select (an initial) folder for the job.
- Folder tag: here you can select a tag to assign to the added folder. You can leave this blank if you do not want to assign a tag to the folder (that is, if all the folders in your job have the same meaning).
- Preview File: here you can select (an initial) file to use as preview for the job.
- **Copy variables in Job**: here you can select if you want to copy all variables from the workable to the Job and where to save them. If you not want to copy and save the variables in the Job, you can leave this parameter blank. If you want to copy and save the variables, you need to set it to the location where you want to save variables (preferably in the custom section of the Job).
- **Define (custom) data to be assigned to the created Job**: here you can set specific data in the new Job (in the database). You can save data in any location in the Job, but you should always save non-standard data in custom sections.

In case you have used a template that also contains (custom) data, the specified data will overwrite any data from the used template, but will be overwritten by direct parameters in this node (for example **Name, Identifier**...).

Select + to add (custom) data. For each entry you need to define:

- A **path**: here you can define a JSON path that selects an element in the JSON document. You need to use a dot notation to reach specific data in the database. Select the pencil to open Expression Builder.
- A value: here you can specify the value you want to set for the specified data path. Select the pencil to open Expression Builder..

See Job data on page 374 for more information.

• Select Created Job: if you select this checkbox, the newly created Job will be active for all the following nodes.

**Form name for 'list edit'**: here you can specify the form you want to use if the Job data need to be displayed and edited in the Job's list view. If you leave this field empty, a form with the name 'Job\_' + <type-of-job> + 'ListEdit' will be searched/used.

Form name for 'form edit': here you can specify the form you want to use if the Job data need to be displayed and edited in the Job's details view. If you leave this field empty, a form with the name 'Job\_' + <type-of-job> + 'FormEdit' will be searched/used.

- Assign this WhitePaper as handler: if you select this checkbox, workables will be created when certain values in the Job change. If you want to select a **Start From Job** on this whitepaper to handle these Job changes, you need to select this checkbox. If you don't want to set or change the handler in the created Job, you need to leave this checkbox unselected.
- Assign this WhitePaper to get forms from: if you select this checkbox, this whitepaper is used to get forms from this whitepaper and to edit them from this whitepaper. If you leave this checkbox unselected, the form setup in the created job is not set or changed from this whitepaper.

# **Create Job**

With this node you can create a new Job.

- **Template Job ID**: here you can enter a unique template Job ID to create a new Job from. The new Job will be a copy of the specified template, copying all predefined information, tasks etc. If you leave this field blank, a completely empty Job is created.
- **Parent Job ID**: here you can enter a unique parent Job ID where the new Job will be nested into. If you leave this field blank, the Job will stand on its own.
- Job Type: here you can select the type of Job. You can use this to store several different sets of Jobs in one database collection. You can leave this blank if your template Job already contains a type.
- **Identifier**: here you can define a (unique) identifier for this Job. This is not an auto assigned value, and there is no constraint on the contents. This value is intended to either provide a back-link to the Job in the master database or to provide a human way to reference the Job in an unique way. The fact that this key is unique is not enforced by CLOUDFLOW. The default value is an empty string.
- Name: here you can define a name of the Job.
- **State**: here you can define the (initial) state of the Job. You can leave this blank if your template Job already contains a type.
- **Description**: here you can define a description for the Job.
- **Team name**: here you can assign a team to the Job.
- Files: here you can selects (an initial) set of files for the Job.
- Files Tag: here you can define a tag to assign to the added files. You can leave this blank if you do not want to assign a tag to the files (that is, if all the files in your Job have the same meaning).
- Folder: here you can select (an initial) folder for the job.
- Folder tag: here you can select a tag to assign to the added folder. You can leave this blank if you do not want to assign a tag to the folder (that is, if all the folders in your job have the same meaning).
- Preview File: here you can select (an initial) file to use as preview for the job.
- **Copy variables in Job**: here you can select if you want to copy all variables from the workable to the Job and where to save them. If you not want to copy and save the variables in the Job, you can leave this parameter blank. If you want to copy and save the variables, you need to set it to the location where you want to save variables (preferably in the custom section of the Job).
- **Define (custom) data to be assigned to the created Job**: here you can set specific data in the new Job (in the database). You can save data in any location in the Job, but you should always save non-standard data in custom sections.

In case you have used a template that also contains (custom) data, the specified data will overwrite any data from the used template, but will be overwritten by direct parameters in this node (for example **Name, Identifier**...).

Select + to add (custom) data. For each entry you need to define:

- A **path**: here you can define a JSON path that selects an element in the JSON document. You need to use a dot notation to reach specific data in the database. Select the pencil to open Expression Builder.
- A value: here you can specify the value you want to set for the specified data path. Select the pencil to open Expression Builder..

See Job data on page 374 for more information.

• Select Created Job: if you select this checkbox, the newly created Job will be active for all the following nodes.

**Form name for 'list edit'**: here you can specify the form you want to use if the Job data need to be displayed and edited in the Job's list view. If you leave this field empty, a form with the name 'Job\_' + <type-of-job> + 'ListEdit' will be searched/used.

Form name for 'form edit': here you can specify the form you want to use if the Job data need to be displayed and edited in the Job's details view. If you leave this field empty, a form with the name 'Job\_' + <type-of-job> + 'FormEdit' will be searched/used.

- Assign this WhitePaper as handler: if you select this checkbox, workables will be created when certain values in the Job change. If you want to select a **Start From Job** on this whitepaper to handle these Job changes, you need to select this checkbox. If you don't want to set or change the handler in the created Job, you need to leave this checkbox unselected.
- Assign this WhitePaper to get forms from: if you select this checkbox, this whitepaper is used to get forms from this whitepaper and to edit them from this whitepaper. If you leave this checkbox unselected, the form setup in the created job is not set or changed from this whitepaper.

### Handle Form

With this node you can build and configure custom Job forms that are used in a Job workflow.

It is used in a Job workflow that contains a Create Job node.

The node has the following configuration options:

- Name: here you can enter the name of the form.
- Form: here you can create the form. See Form Builder on page 260 for more information on how to do this.

#### Next Job Phase

With this node you can set the next Job states and set the workable on hold here until the state of the linked Job has changed.

This node is a combination of the nodes Update Job, Wait for Job State Change and Route on Job State.

The node checks job states on start, and if it matches one of the next states, then it will continue without waiting.

It has the following configuration options:

- Next states: here you can define the next Job states. For every state, an output connector is created for the node.
- Also allow other states: if you select this checkbox, Job states that are not specified are also allowed.

# Select Job

With this node you can select an active Job in the workflow that will be used for all the following nodes.

It has the following configuration options:

- Select By: here you can specify how you want to choose, find or select the Job. Options:
  - Job ID: if you select this option, the Job that has the specified Job ID is selected.
  - Job Name: if you select this option, the (first) Job that has the specified name is selected.
  - Job Identifier: if you select this option, the (first) Job that has the specified identifier is selected.
  - File: if you select this option, the (first) Job that contains a file that matches the specified URL is selected. You can also specify a tag to limit the URLs that can match.
  - Folder: if you select this option, the (first) Job that contains a folder that matches the specified URL is selected. You can also specify a tag to limit the URLs that can match.

# If Select By is set to Job ID

• Job ID: here you can specify the Job ID.

### If Select By is set to Job Name

• Job Name: here you can specify the Job name.

### If Select By is set to Job Identifier

• Job ID: here you can specify the Job identifier.

# If Select By is set to File

- File or Folder: here you can specify the URL of the file or folder that should be used to find the Job.
- **Tag**: here you can specify the tag of the file or folder that is used to find the Job. If you leave this field blank, all files and/or folders of a Job will be searched. If you do not leave this field blank, only the files and/or folders that have a matching tag will be used.

# If Select By is set to Folder

- File or Folder: here you can specify the URL of the file or folder that should be used to find the Job.
- **Tag**: here you can specify the tag of the file or folder that is used to find the Job. If you leave this field blank, all files and/or folders of a Job will be searched. If you do not leave this field blank, only the files and/or folders that have a matching tag will be used.
- Any enclosing folder: with this checkbox you can specify if the folder URL needs to match completely or partially. If you leave this checkbox unselected, the URL exactly needs to match a folder in a Job (the Job that will be selected). If you select this checkbox, you can select a Job that has a folder that encloses the specified URL. For example, if you specify the full file URL, a Job that just matches the beginning of the URL is selected.

# Start From Job

With this node you can start a workflow when a certain value in a Job changes.

It has the following configuration options:

- Handle State Change: if you select this checkbox, a flow will be started when the state of a Job changes. The following variables will be created:
  - event: StateChanged
  - **old\_state**: the old state
  - state: the (new) state
- Workable name: here you can specify the name to assign to the created workable. Leave this parameter empty if you do not have a specific name to assign. Select the pencil to open Expression Builder.
- Link to Jacket (by name): here you can specify a jacket with the name that is specified here will be searched. If found, the new workable will be linked to the jacket. If not found, a new jacket will be created, and the specified name will be assigned to it. Leave this parameter empty if you do not want to link to an existing jacket. Select the pencil to open Expression Builder.
- **Priority**: in this drop-down list you can specify the priority to assign to the created workable.

# **Update Job**

With this node you can update the information in a Job.

- Activity: here you can select the information to change. Options:
  - Add Files: if you select this option, you can add new files to the Job.
  - Add Folder: if you select this option, you can add a new folder to the Job.
  - Add Versions: if you select this option, you can add new versions of the file(s) to the Job.
  - **Remove Files**: if you select this option, you can remove files from the Job.
  - **Remove Folder**: if you select this option, you can remove a folder from the Job. This option does not affect the folder on disk.
  - Move Files: if you select this option, you can move or rename files in the Job. This option does not affect the files on disk.
  - **Move Folder**: if you select this option, you can move or rename a folder in the Job. This option does not affect the folder on disk.
  - Set Next States: if you select this option, you can define the possible next states for the Job.
  - Set State: if you select this option, you can set the state of the Job.
  - Set Data: if you select this option, you can set the data of the Job.
  - Set Preview: if you select this option, you can assign a preview to the Job.
  - Set Form Name: if you select this option, you can change the Job form.
  - Link to sub job: if you select this option, you can link a sub Job to the Job.
  - Unlink to sub job: if you select this option, you can unlink a sub Job from the Job.
  - Mark final: if you select this option, you can mark the Job as final. A final Job can no longer be edited.
- Job: here you can select a Job to update. In most cases you can leave this field blank because in most cases you don't need to explicitly select a Job. For example, a Job will be linked to this workable if this workable passed a **Create Job** node or if the workable was created from a **Start From Job** node.

However, in some cases you want to update (another) Job or a Job from different setups. In that case you can select this Job with this parameter. There are two ways to select the Job:

- **By Job ID**: you can specify here the Job ID of the Job you want to update. This is the ID that is automatically assigned to the Job by CLOUDFLOW when you create the Job.
- By Identifier: you can also specify the Identifier you assigned yourself to the Job when you created it.

CLOUDFLOW will automatically recognize the method you are using, and load the correct Job.

# In case Activity is set to Add Files

- Files: here you can select the (set) of files to add to the Job.
- Files Tag: here you can specify a tag to assign to the added files. In case all the files in your Job have the same meaning you can leave this blank.
- Assign Tag from enclosing job folder: if you select this checkbox, the tag of the enclosing folder is automatically assigned to the added file. First a job folder is searched that encloses the new file. Then the tag assigned to that folder is used to assign as tag to the new added file. If more than one folder encloses the file, the folder that does not enclose any of the other enclosing folders is selected. If no enclosing folder is found, no tag will be assigned. This option overrules the tag specified in the parameter Files Tag.
- Reject files outside job folder: if you select this checkbox, files that are outside a Job folder will be rejected.
- Update Existing: if you select this checkbox, existing files in the Job are replaced if it matches the specified URL(s). If you don't select this checkbox, files are added (and not replaced), even if the specified URL(s) already exist in the Job.

# In case Activity is set to Add Folder

- Folder: here you can select the folder to add to the Job.
- Folder Tag: here you can specify a tag to assign to the added folder. In case all the folders in your Job have the same meaning you can leave this blank.
- Update Existing: if you select this checkbox, existing folders in the Job are replaced if it matches the specified URL(s). If you don't select this checkbox, folders are added (and not replaced), even if the specified URL(s) already exist in the Job.
## In case Activity is set to Add Versions

- Files: here you can select the (set) of files to add to the Job.
- Files Tag: here you can specify a tag to assign to the added files. In case all the files in your Job have the same meaning you can leave this blank.
- **Previous Versions of Files**: here you can select the previous versions of the file (**Files** specifies the new version of the file). Make sure to set **Previous Versions of Files** to the URL of the previous file the way it is named on disk and **Files** to the URL of the current/new file the way it is named on disk.

If:	Then:
The previous file version still has the same name and the new file version has a new name.	<ul> <li>The file that corresponds to the specified previous version will be searched.</li> <li>The versions in that entry will be updated.</li> <li>The references in that entry will be changed to the new URL.</li> </ul>
The previous file version has been moved/renamed and the new file version has same name as the before.	<ul> <li>The file that corresponds to the specified new version will be searched.</li> <li>The versions in that entry will be updated.</li> </ul>

Which one of these two modes is used is determined automatically. However, it is the users responsibility to organize the previous and new file correctly on the file system before calling the node.

#### In case Activity is set to Remove Files

- Files: here you can select the (set) of files to remove from the Job.
- Files Tag: here you can specify a tag to assign to the removed files. In case all the files in your Job have the same meaning you can leave this blank.

## In case Activity is set to Remove Folder

- Folder: here you can select the folder to remove from the Job.
- Folder Tag: here you can specify a tag to assign to the removed folder. In case all the folders in your Job have the same meaning you can leave this blank.

#### In case Activity is set to Move Files

• Previous Location of Files: here you can specify the location where the files used to be located.

#### In case Activity is set to Move Folder

- Previous Location of Folder: here you can specify the location where the folder used to be located.
- Also update all files inside the moved folder: if you select this checkbox, the files inside a moved folder are also moved. If you leave this checkbox unselected, the files inside a moved folder are not moved and you have to move the files yourself. This can for example be the case if you want to keep the references to the file(s).

#### In case Activity is set to Set Next States

- **Possible Next States**: if you select +, you can define possible next states. Select the pencil to open Expression Builder. The possible next states are visible in the **DATA** tab in the Job details. If you select one of the states, this state will be set to the Job.
- Also allow other states: if you select this checkbox, other states that are not specified in Possible next States are also allowed. This can be needed for example when a job should be reset to an earlier state if in case of a problem.

#### In case Activity is set to Set State

- State: here you can specify the state to assign to the Job.
- Force Events: if you select this checkbox, workables are created in the Start From Job node, even if changes are not actually applied. Normally, no workable will be created when you use Set State and when the new value for the state is the same as the current value. However, if you select this option, a workable will be created in all cases.

#### In case Activity is set to Set Data

With this option you can set or update specific Job data in the database. You can save data in any location in the Job, but you should always save non-standard data in custom sections.

In case you have used a template that also contains (custom) data, the specified data will overwrite any data from the used template, but will be overwritten by direct parameters in this node (for example **Name**, **Identifier**...).

Select + to specify the path and the value of the data you want to set or update.

- **Path** : the path to the data. You need to use a dot notation to reach specific data in the database.
- Value: the value you want to set or add to the specified data path.

#### Example

Path	Value
custom.cusname	CustomerX
custom.cusID	12345
identifier	Job_442542

This will result in the following JSON file in the database:

```
{
    "identifier" : "Job_442542",
    "custom" : {
        "cusname" : "CustomerX",
        "cusID" : "12345",
    }
}
```

- In case the fields **cusname** or **cusID** do not exist, they will be created.
- In case the fields do exist but contain different values, the values of the fields will be updated.

#### In case Activity is set to Set Preview

• **Preview file**: here you can select a file to use as preview for the Job.

#### In case Activity is set to Set Form Name

- Form Type: here you can select the form type. Options:
  - List Edit: if you select this option, the form Job data are displayed and can be edited in the Job's list view.
  - Form Edit: if you select this option, the form Job data are displayed and can be edited in the Job's details view.
- Form Name: here you can specify the form you want to use. See Handle Form on page 250 for more information on Job forms.

#### In case Activity is set to Link to sub job

- Project ID: here you can define the project ID of the sub Job to link to this Job to.
- Project Tag: here you can define an optional tag to assign to the linked sub Job.

## In case Activity is set to Unlink sub job

- Project ID: here you can define the project ID of the sub Job to unlink to this Job from.
- **Project Tag**: here you can define an optional tag to assign to the unlinked sub Job.

## In case Activity is set to Mark final

• Make Final: if you select this checkbox, the Job is marked as final. In this case, the Job can not be edited anymore.

## Wait For Job State Change

This node will stop the workable until the state of the linked Job changes.

# Mail

## Send E-mail

With this node you can send an e-mail.

It has the following configuration options:

- From: here you can enter the email address of the sender.
- To: here you can enter the email addresses of the recipients. Select + to add a recipient.
- CC: here you can enter the email addresses of the recipients in CC. Select + to add a recipient.
- BCC: here you can enter the email addresses of the recipients in BCC. Select + to add a recipient.
- Mail Data Source: here you can specify whether the body and subject data are specified in the parameters of this node, or if they are coming from a template. Options:
  - From Parameters: if you select this option, the subject and body can be specified in the node.
    - Subject: here you can enter the subject line. Select the pencil to open Expression Builder.
    - Body: here you can specify the e-mail body. Depending on the Mail Body Type parameter, this can be:
      - As is (when the **Body Type** is not template based): in this case, the specified text is used for the body without any additional processing. Select the pencil to open Expression Builder.
      - A **template** (when the **Body Type** is template based): in this case, the specified text is used as input for template processing. All sections starting with \${ and ending with } will be searched and replaced by contents of the *workable*, *jacket* etc. The following replacements will be performed, depending on the begin of the string after \${
        - **variable**: this will look up a variable that has the same name as the remaining part. For example **\${variable.test}** will be replaced by the value of a variable called **test**.

Select the pencil to open Expression Builder.

- From Template: if you select this option, the subject and body from a specified template will be used.
  - **Template Name**: here you can enter the name of the template that needs to be used for the subject and the body. This name will be used in combination with a language to select the template.
  - **Template Language From User**: if you leave this checkbox unselected, the language to be used to select the correct template is specified by the **Template Language** parameter. If you select this checkbox, the language of the user will be used. By default the language specified in the **Template Language** will be used if no template can be found that has the specified name and the language of the user.
  - **Template Language**: here you can specify the expected language for the template. This parameter will be used in combination with the **Template Name** parameter to locate the actual template. If no match is found with the specified name and language, the language **english** will be used. If not found, the first template with the specified name will be used, ignoring the language.

- Mail body type: here you can enter the system to be used in the Body parameter. Options:
  - Plain: if you select this option, the body text will be used as is to send a plain text e-mail.
  - HTML: if you select this option, the body text will be used as is to send an HTML e-mail.
  - **Plain+Template**: if you select this option, the body text will be used after template processing to send a plain text e-mail.
  - **HTML+Template**: if you select this option, the body text will be used after template processing to send an HTML e-mail.

When using template mode, the following parameters can be used:

• **\${variable.<name>}**: this will be replaced by the value of the variable **name**.

If you want to reference on of the previews that you might have specified, you could for example use <img src='cid:preview-1'> to reference the first preview, <img src='cid:preview-2'> to reference the second preview...

- Attachments: here you can enter the files to attach. Leave this parameter empty if you do not want to attach any files.
- **Previews**: here you can specify the files that can be used to add previews in the HTML mail body. If you do not want to add previews you can set this to an empty array or empty string.
- **Number of retries**: here you can specify the number of times the node should retry to send. 0 means no retries. The system will only attempt to retry when the server could not be reached. All other cases (for example wrong authentication) will not trigger a retry.
- Time to wait between retries: here you can specify how long the node should wait before retrying.

## 3D

## Render 3D View To File

With this node you can render a view of an IC3D file and save it.

It has the following configuration options:

- Input File: here you can specify the files where you want to render a view from. Select the pencil to open Expression Builder.
- View Yaw: here you can specify the yaw of the view you want to render.
- View Pitch: here you can specify the pitch of the view you want to render.
- **Output File**: here you can specify he result names for the generate view file(s). Options:
  - If you leave this field blank, an output file with the original name with the correct extension will be written in the same folder as the input file.
  - If you define an **absolute CLOUDFLOW URL containing a filename** (cloudflow://..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you define an absolute CLOUDFLOW URL without a filename (cloudflow: //..., see Server URL for more information), an output file with the original file name will be written in the specified folder.
  - If you enter a **relative path containing a file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
  - If you enter a **relative path without a file name**, an output file with the original file name will be written in the specified folder based on the original input folder.

You can specify one value or as many files as you have specified for the Input Files.

- If you specify one value, this value will be combined with each input file to generate an output file.
- If you specify as many files as you have specified for the Input Files, each value will be combined with each input file to generate an output file.
- Image Type: here you can specify the image format to use when saving the view. Options:
  - JPEG.
  - PNG. PNG files will be saved with a transparent background.

- Render Type: here you can specify how the 3D file should be rendered. Options:
  - Standard: if you select this option, standard 3D rendering is used.
  - Ray Tracing: if you select this option, ray-tracing for rendering is used.
- Nr Samples (only available in case of Ray Tracing: each pixel in an image that is generated by ray tracing is built from several rays. Here you can specify the number of rays that will be used for one pixel.
- Resolution: here you can specify the resolution of the rendered view.
- Width: here you can specify the width of the rendered view.
- Height: here you can specify the height of the rendered view.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

## **Replace Labels in IC3D**

With this node you can replace one or more labels in an IC3D file.

It has the following configuration options:

- IC3D File: here you can specifies the input IC3D file. Select the pencil to open Expression Builder.
- **Replace Label**: here you can specify the name of the label in the IC3D file where you want to assign a new PDF file to.
- **Replace Back Side of Label**: if you select this checkbox, the back side of the label will be replaced. If you leave this checkbox unselected, the front side of the label will be replaced.
- Assign PDF File: here you can specify the URL of the PDF file to assign to the label.
- Assign Page: here you can specify the page of the PDF file to assign to the label. Set it to zero in case you want to get the first page of the PDF file or in case you want to assign a single page PDF file.
- **Resolution**: here you can specify the resolution to use when rendering the newline assigned PDF file.
- Output IC3D File: here you can specify an alternative name to save the updated IC3D file. Options:
  - If you leave this field blank and if the checkbox **Overwrite existing file** is selected, the file will be overwritten.
  - If you define an **absolute CLOUDFLOW URL containing an alternative filename** (cloudflow://..., see Server URL for more information), an output file with the specified file name will be written in the specified folder.
  - If you enter a **relative path containing an alternative file name**, an output file with the specified file name will be written in the specified folder based on the original input folder.
- **Overwrite existing file**: if you select this checkbox, files that already exist will be overwritten. If you leave this checkbox unselected, existing files will not be overwritten and an error is displayed.

#### Update Labels in IC3D

With this node you can update and/or render labels in a IC3D file.

It has the following configuration options:

- IC3D File: here you can specify the input IC3D file. Select the pencil to open Expression Builder.
- **Resolution**: here you can specify the resolution to use when rendering.
- Render Updated Only: if you select this checkbox, only the updated labels will be (re)rendered. If you leave this checkbox unselected, all labels will be (re)rendered.

## Presets

With these nodes you can use and configure the Standalone RIP Preset in a workflow.

For each preset you create in the Standalone RIP, a Preset node is generated.

You can configure and use the Preset nodes in a workflow.



**Note:** Configuring a Preset node in a workflow does not change the preset in the standalone RIP. You can consider the RIP Presets as a template which can be further configured via the corresponding node in a workflow.

For more information on Standalone RIP Presets and how to create them, see Presets on page 350 for more information.

# KIOSK

With KIOSK you can interact with existing workflows and submit a job to a workflow.

It is a separate user interface for people who are only using workflows rather than building them (for example operators).

- You can submit a job to a workflow.
- It provides an overview of the jackets.

The KIOSK tab in the CLOUDFLOW navigation bar displays a badge number with all the pending workables where the Handler correspond to the current user.

# Submit a job to a workflow

In KIOSK you can submit a job to a workflow.

To do this, follow these steps:

- 1. Select Create New.
- 2. Select the Workflow you want to submit to fro the drop-down list.

**Note:** You can mark the workflow as favorite by selecting the star icon at the right side of the workflow in the drop-down list. Favorite workflows appear on top of the list.

- 3. Select the Input.
- 4. If needed, provide extra parameters for the workflow. See Form Builder on page 260 for more information on how to predefine these parameters.
- 5. Browse to the file(s) to upload them or select them from a File store. See Enable File store browsing in the Start from KIOSK node for more information on file uploading.
- 6. Select Submit.

When a job is submitted to a workflow, a jacket with at least one workable is created. See Jacket on page 472 and Workable on page 474 for more information.

## Jackets overview

KIOSK provides you with an overview of the jackets.

## Filter jackets

In the left panel in KIOSK you can see an overview of all the jackets. A jacket, containing at least one workable, is generated when a job is submitted to a workflow. See Jacket on page 472 and Workable on page 474 for more information.

You can use the following filter options:

- Search field: if you enter a query in this field, all jackets containing that query will be displayed. In case of Jobs, you can search on the Job's name, status, description and the name of the input files.
- Error: if you select this filter, only the jackets with an error status are displayed.
- Hold: if you select this filter, only the jackets that are on hold are displayed (for example in case of approval flows where files need to be assessed before continuing).
- **Running**: if you select this filter, only the running jackets that are displayed.
- Finished: if you select this filter, only the finished jackets are displayed.
- Modified/Created: if you select this filter, only the jackets that are created or modified at a specified moment are displayed.
- Workflow: if you select this filter, only the jackets that were submitted to a specific workflow on are displayed.

#### • User

- All Users: if you select this filter, all jackets of any user are displayed.
- Just Me: if you select this filter, only the jackets of the current user are displayed.

Save a filter and apply a saved filter:

- To save a filter, select + Save filter.
- To select a saved filter and apply it, select **T**.

If you select a jacket, you can:

- **Reload** the jacket (C). You can relaunch the jacket with the same original files and parameters, or you can select other files and parameters. To select other files, select **x** next to **Same original Files** and browse to the file(s) to upload them or select them from a File store.
- Cancel jackets that are on hold (<sup>1</sup>).
- Delete the jacket (1). This is only possible when the flow has been finished.

If you select a jacket, you can select one of the following display options:

# Overview

In **Overview** you can see an overview of the *workables* that are in the *jacket*. A jacket can contain one or more workables:

## Example

1		• 400488	9/26/2016, 3:53:22 PM C	1	<b>₩ 400498</b>			
<b> </b> ~		400500	9/26/2016, 3:37:46 PM			Start	ID6-Accurate - Start From Klosk	
<b>)</b> ~		400500	9/26/2016, 3:36:28 PM			Current State Finished	ID6-Approve - End normal yes	
*		400497	9/26/2016, 3:31:01 PM		A. Bitter JE.			
×	2	Demo design	9/26/2016, 3:26:11 PM					В
			I <sup>MI</sup> 400498 - cloudfiow_admin@hybridsoftware.com				-	
					The second	Start Current State Finished	ID6-Approve - Start Approval ID6-Approve - Route After Approval normal yes	

- A: one jacket
- **B**: two workables

In case of an approval cycle, you can approve or reject a file with the decision buttons. If you are a user with administrator rights, you can force the file to be approved are rejected.

# Log

In Log you can see a detailed report of the activity of the selected *jacket*.

Each *workable* has its own track. You can see the different nodes that were passed and how much time the processing took.

# Analyse 🛄

In Analyse you can download the selected jacket.



Note: Only administrators can download a jacket.

# Jobs 🗏

In **Jobs** you can see the Job details, in case you're using the JOBS module. See JOBS on page 372 for more information about this module.

Depending on what you have defined in **SETTINGS** > **SETTINGS** > **JOBS**, the **Details link** refers to a predefined Job link. See JOBS on page 50 for more information.

# Access KIOSK

Next to the KIOSK tab, there are various ways to reach KIOSK.

- Via the **Play** button in **Workflow Editor**.
- Via the WORKFLOW tab in the ASSET tab.
- Via the **Submit** option in **PACKZ**. See the PACKZ reference manual for more information.

You can also use URL parameters for filtering in KIOSK. The following parameters are possible:

- searchText
- =Finished
- =Error
- =Hold
- =Running
- beginDate
- endDate

#### Example

If you add &whitepaper\_name=Preflight&state=Finished to the KIOSK URL, KIOSK will filter on all finished workflows called **Preflight**.

# Form Builder

With Form Builder you can create custom forms.

A custom form contains fields in which the user sees information or which the user needs to fill in.

#### Examples of cases when a form is or can be displayed

- When submitting a job to a workflow with **KIOSK**. See Submit a job to a workflow on page 258 for more information.
- To release a job in KIOSK that was on hold.
- When creating a new Job. See JOBS on page 372 and Create Job on page 248 for more information.
- When adding information to the Job information form in the Jobs overview or to the Job details form.
- When approving a file in PROOFSCOPE.

You can build a custom form in the following nodes:

- Start from Kiosk
- Hold in Kiosk
- Handle Form
- Start Approval

When a workflow which contains one of these nodes is used, the user sees information or needs to fill in the fields of this form.

# How to build a form

This chapter gives an overview of the steps you need to take to build a form.

To build the form, follow these steps:

- 1. Open one of the three nodes where you can build a custom form.
- 2. Select the pencil next to Kiosk parameters/Form.
- **3.** The window opens where you can configure the form fields. On the top right, select the field type you want to add. Options (see Overview of the KIOSK parameters on page 261 for more info or select a field type to see its specific configuration settings):
  - **a.** Text: with these settings you can create a text field.
  - **b.** Paragraph: with these settings you can create a paragraph field.
  - c. Checkbox: with these settings you can create a checkbox.
  - d. Number: with these settings you can create a number field.
  - e. Number with decimals: with these settings you can create a number field which allows decimals.
  - **f.** Choose from a list: with these settings you can create a field containing a drop-down list from which the user can select one item.
  - **g.** Multi select from a file list: with these settings you can create a field containing a selection list from which the user can select one or more items.
  - h. Unit: with these settings you can create a field containing a unit.
  - i. Date: with these settings you can create a field containing a date.
  - j. Interactive lists: with these settings you can create interactive lists.
  - **k.** Auto Fill: with these settings you can generate auto fill suggestions from custom object collections in the database.
  - **I.** Rip Settings: with these settings you can create a field containing RIP settings. See RIP for more information on RIP settings.
  - **m.** Title: with these settings you can create a title or subtitle in the web form.
  - **n.** Image: with these settings you can create an image field.
  - **o.** Error message: with these settings you can create a field containing an error message.
- 4. The fields appear on the left side of the screen.
  - To add a field, hover over it and select + Add new field.
  - To delete a field, hover over it and select 💼.
  - To move a field, select 🔀 on the left.
- 5. Order the fields on the grid. There are four columns in a form, you can move and resize fields according to the grid.
  - To extend the grid, select +.
  - To delete a grid field, select 🖻.

## **Overview of the KIOSK parameters**

This is an overview of the KIOSK parameters that you can configure when creating a form.

## Text

With **Text** you can add text. The value of the text will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.
- **Default**: the default value of the field.

- Text in background: the background text.
- Pattern: a Regular expressions on page 475 to force the user to use a specific pattern for the text.
  - **Note:** You can use https://regex101.com as an online regex tester.
  - **Note:** If the KIOSK user uses an expression that doesn't match the pattern, an error is displayed.
- Pattern Error Message: the error that will be displayed when the text does not correspond to the pattern.

## Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### **Form Actions**

With Form Actions you can show or hide a field, depending on the value of another specified field.

#### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

## Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

When

• Is equal to hide

## Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

## Paragraph

With Paragraph you can add a paragraph. The value of the paragraph will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.

#### • Field options:

- **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- Readonly: if you select this checkbox, the field is read only and cannot be filled in.

## Value Options

- **Default**: the default value of the field.
- Text in background: the background text.
- Number of rows: the number of rows in the paragraph.

### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### Form Actions

With Form Actions you can show or hide a field, depending on the value of another specified field.

#### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

## Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

#### When

• Is equal to hide

#### Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

## Checkbox

With Checkbox you can add checkboxes. The value of the checkboxes will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.

- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - **Readonly**: if you select this checkbox, the field is read only and cannot be filled in.

## Value Options

- **Default**: the default status of the checkbox.
- Label checkbox: the label of the checkbox.

## Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

#### Form Actions

With Form Actions you can show or hide a field, depending on the value of another specified field.

#### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

## Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

#### When

• Is equal to hide

#### Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

## Number - Number with decimals

With Number you can add numbers or numbers with decimals. The value of the numbers will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.

- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

## Value Options

- Default number: the default number.
- Minimum: the minimum number.
- Maximum: the maximum number.

## Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

## Form Actions

With Form Actions you can show or hide a field, depending on the value of another specified field.

## When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

#### Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

## Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

When

• Is equal to hide

## Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

## Choose from a list

With **Choose from a list** you can add a list with options. The user needs to choose one of the options from a dropdown list. The value of the user's choice will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object.

For example: **customer.name**, **customer.address** and **customer.telephone** will result in one variable **customer** that is an object with three fields **name**, **address** and **telephone**.

- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

#### Value Options

- **Default**: the default option.
- List possibilities: the options in the drop-down list. Select 🕒 to add an option.
- **Output format**: the output format.

#### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### **Form Actions**

With Form Actions you can show or hide a field, depending on the value of another specified field.

#### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

## Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

When

• Is equal to hide

Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

#### Multi select from a file list

With **Multi select from a file list** you can add a list with options. The user can select one or more of the options in a checklist. The value of the user's choice will be saved in a variable array.

• **Type**: the type of the parameter.

- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.

## Value Options

- Entire List: the options in the list. To enter multiple options, use this format: ["option1", "option2", "option3"...].
- Selected items: the preselected options. These options must be available in the entire list.

## Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

## Unit

With Unit you can add a unit. The value of the unit will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

## **Value Options**

- **Default number**: the default number.
- Unit type: the type of the unit.
- Storage unit: the unit that is used to store the value in the variable. For example, if the Unit type is length, the Storage unit is in and the user enters 10 in the form, the variable value will be 0.393701.
- Minimum: the minimum value of the unit.
- **Maximum**: the maximum value of the unit.

## **Note:** Unit preferences

The units displayed in the node's UI will be those set in CLOUDFLOW's general user preferences. In case of conflicts these fall back on the system preferences.

## Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

## Date

With **Date** you can add a date. The value of the date will be saved in a variable. The format of the date is defined by your browser language.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. You can use dot notation if you want to group single variables into a single object. The variable will be addressed in the workflow with this name. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- **Help text**: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

## Value Options

- **Default** : the default date. By default, **Always current date** is selected. In this case, the current date is always selected. Select **X** to choose a specific date.
- **Display format**: the display format.

## Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

## Radio buttons

With **Radio Buttons** you can add radio buttons. The user needs to select one of the radio buttons. The value of the selected radio button will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required?**: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.

## Value Options

- Url of image: the URL to browse to an image you want to display next to the radio button. The image needs to be available on a webserver.
- Text: the text that will appear next to the radio button.
- Value: the value of the radio button.
- **1**: to add a radio button.
- 🔲: to delete a radio button.
- Width (px): the width (in pixels) of the image.
- Height (px): the height (in pixels) of the image.
- **Default value**: the value you want to set as default value.

## Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

## **Form Actions**

With Form Actions you can show or hide a field, depending on the value of another specified field.

### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

#### Then

- Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.
- Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

#### When

• Is equal to hide

#### Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

#### Interactive lists

With **Interactive lists** you can add interactive lists. The user needs to select an option from a drop-down list, where the options in the second list depend on the selected option of the first list. The values of the user's selection will be saved in a variable.

- **Type**: the type of the parameter.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

## Value Options

- First Label: the first label in the KIOSK interface.
- First key: the name by which the variable will be addressed in the workflow.
- **Output format**: the output format.
- Second Label: the second label in the KIOSK interface.
- Second key: the name by which the variable will be addressed in the workflow.
- **Output format**: the output format.

Values: the values in the first and second labels. Select 🗄 to add a value.

- The values of the first label are the ones without an indent.
- The values of the second label are the ones with an indent.

The values of the second label depend on the selected values of the first label.

#### Example

The following setup implies that the user can select two Areas from a drop-down list: Area X and Area Y. Depending on the selected Area, the user can select a Company:

- Company 1 or Company 2 in case Area X was selected.
- Company 3 or Company 4 in case Area Y was selected.

Values*	Area X
	Company 1
	Company 2
	Area Y
	Company 3
	Company 4

The drop-down lists look like this in KIOSK:

Input	Start flow	/	÷
Options			
	Area	Area X	\$

#### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### Readonly

With **Readonly** you can add a field with a text value.

- Type: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.

## **Value Options**

- **Default**: the default value of the field.
- Text Style: the style of the text.

## Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

## Autofill

With Auto Fill you can to create a field with auto fill suggestions coming from the users and custom object collections in the database.

When you start entering content in the field, all the suggestions that match the value are visible and selectable. The values of the field will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- Collection: the collection in the database you want to retrieve values from. Options:
  - Users: if you select this option, the suggestions are retrieved from the users collection in the database.
  - **Custom objects**: if you select this option, the suggestions are retrieved from the customs object collection in the database.
  - **Dataconnector**: if you select this option, the suggestions are retrieved from an external data source (for example DATALINK).
- Custom object name (only in case of Custom objects): the name of the custom object. For example, if you have a database collection customobjects.customers, the custom object name is customers.
- Connector (only in case of Dataconnector): here you can specify the connector.
- **Table name** (only in case of **Dataconnector**): here you can specify the name of the table you want to use to retrieve data for auto fill suggestions.
- Key field: the column name of the field of which the content will be written to the variable.
- Label field: the column name of the label that will be displayed when you start typing.
- Required
  - Value must exist: if you select this checkbox, the value must exist in the collection where Auto Fill is linked to. If you leave this checkbox unselected, you can enter a value that does not exist. For example, when administrating customer collections, you can add a new customer directly without having the add the customer to the database first.
- Examples

Example of a users database collection:

fullname	email	username
Administrator	admin@company.com	admin
Jane	Jane@company.com	jane
John	John@company.com	john

fullname	email	username
June	June@company.com	june
Jen	Jen@company.com	jen

If you configure the form like this:

## Key field: email and Label field: fullname

The field in KIOSK will retrieve the suggestions from the field **fullname** and will write the value of the field **email** to the variable. For example, if you enter **e**, suggestions **Jane**, **June** and **Jen** are shown (because they all contain **e**. If you select **June**, the resulting variable will have value **june@company.com**.

- Maximum amount to show: the maximum amount of characters that will be shown.
- Min length: the minimum length from which the auto fill suggestions will be shown.
- Delay (ms): the delay before the auto fill suggestions will be shown.
- Multiple: if you select this checkbox, you can enter multiple items in the Auto Fill field.

#### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### Make list

With Make list you can create a list with input fields. The value of the fields will be saved in a variable.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- Default list: here you can create the list input fields. Select 단 to add an input field.

#### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

### **Form Actions**

With Form Actions you can show or hide a field, depending on the value of another specified field.

#### When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

## Then

• Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.

• Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

#### Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

## When

• Is equal to hide

#### Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

#### Item repeater

With Item repeater you can display the items of a variable list of an array and edit them in the output variable. **Item repeater** is only available in **Hold In Kiosk**.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- Default: the default value. In this case it is the variable array where you want to repeat the items from.

#### Value Options

- Edit: the button to enter the window where you can edit the parameters of the array you want to display in your form.
  - **Type**: the type of the parameter.
  - (variable) key: the name of the variable that contains the parameter value. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.

```
If you use an array of objects, it should be like this example: [{value: 8, name: "first"},
{value: 10, name: "second"}, {value: 12, name: "third"}, {value: 14, name:
```

"last" }]. In the item settings, you need to define the parameter of the object you want to use, for example "value" or "name".

- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required?**: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
  - Readonly: if you select this checkbox, the field is read only and cannot be filled in.

#### **Value Options**

- **Default**: the default value.
- Text in background: the background text.
- Pattern: a Regular expressions on page 475 to force the user to use a specific pattern for the text.
  - **Note:** You can use https://regex101.com as an online regex tester.
  - **Note:** If the KIOSK user uses an expression that doesn't match the pattern, an error is displayed.
- Pattern Error Message: the error that will be displayed when the text does not correspond to the pattern.

Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

Select Go back to return to the general parameters of Item repeater.

#### Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

#### Example

You want to extract the separations of a file and display tinthem in a form in **Hold in Kiosk**. If needed, the user can change the name of the separations.

This is the workflow:



In Get Meta Data, the Variable name parameter is set to metadata. Consequently, if you run the workflow, the output of Get Meta Data would be something like this:

LOGS	VARIABLES	OUTPUT	RES-PARAMS	EXE-PARAMS	ADD BREAK
LOGS output [ { pa va	VARIABLES ass: 1 iriables: { metadata { number_of output_colo type: "su colorant 0: { na typ	OUTPUT	ges: 1	EXE-PARAMS	ADD BREAK
		0:0			

This means that CLOUDFLOW needs the following notation to extract the separations from the array in the output variable: **\${metadata.output\_color\_space.colorants}**.

To configure the form, follow these steps:

- 1. Open the parameters in Hold in Kiosk.
- 2. Select the pencil next to Kiosk parameters.
- 3. Select Add new field and select Item repeater from the drop-down list.
- 4. In Value Options > Default, enter \${metadata.output\_color\_space.colorants} to extract the variables in colorants from the output array.
- 5. Select Edit to edit the parameters.
- 6. In (variable) Key, enter name, since this is the variable in the array we need to address to extract the name of the separations.

LOGS	VARIABLES	OUTPUT	RES-PARAMS	EXE-PARAMS	ADD BREAK
output [ { va	ass: 1 rriables: { metadata: { number_of number_of output_color type: "su colorants 0: { na typ	_pages: 1 _logical_pag or_space: { ubtractive" s: [ me:]"Black" e: "standard eviews: { cmyk: [ 0: 0 1: 0	ges: 1 d"		

- 7. If needed, configure other parameter values such as Label, Help text, Text in background ...
- 8. Select Save to save the form.
- 9. Select Save to save the parameters in Hold In Kiosk.

10. If you run the workflow, the result in the form in Hold In Kiosk would be something like this:

■ 87654				
	Start Current State Finished	1 test - Start From Kiosk 1 test - Hold In Kiosk normal no		
OPTIONS				
		Labeltext	Color	Black
			Color	Magenta
			Color	Yellow
			Color	Cyan
			Color	keyline
			Color	PANTONE 871 C
			Color	PANTONE 137 C
			Color	PACKZ INFOPANEL
			Decision	

11. If needed, the user can change the name of one or more the color. This will be saved in the output variable of Hold In Kiosk. For example, if the user changes keyline to diecut, this would be the result:



# **Rip Settings**

=

With **RIP** Settings you can create form entries that contain the same RIP input fields that exist in the RIP node or standalone RIP. The value of the fields will be saved in a variable.

Note: In the left window, the RIP settings will appear. See RIP for more information.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required**?: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- **Default preset**: the default preset.

#### Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

#### Auto fill by table

With **Auto fill by table** you can look up information from a table and automatically fill in several text fields in the form with this information.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. This key needs to correspond to the key of the table column that you want to address to fill in the values.
- Label: the label in the KIOSK interface.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required?**: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- **Default**: the default value of the field.
- Text in background: the background text.
- Collection: the collection in the database you want to retrieve values from. Options:
  - **Custom objects**: if you select this option, the suggestions are retrieved from the customs object collection in the database.
  - **Dataconnector**: if you select this option, the suggestions are retrieved from an external data source (for example DATALINK).
  - **Jobs**: if you select this option, the suggestions are retrieved from the Jobs collection in the database. You can use dot notation.
  - Start From DBIO: if you select this option, the suggestions are retrieved from the data that is collected from a workable that was created by a workflow that starts with the Start from DBIO and sets a reply with the Set DBIO Reply.
- **Custom object name** (only in case of **Custom objects**): the name of the custom object. For example, if you have a database collection **customobjects.customers**, the custom object name is **customers**.
- Connector (only in case of Dataconnector): here you can specify the connector.
- **Table name** (only in case of **Dataconnector**): here you can specify the name of the table you want to use to retrieve data for auto fill suggestions.

- Columns: the columns you want to add to the table to retrieve data from. Select 🖋 to open the Manage columns tab.
  - Select + or Add column to add a column.
  - Select 🕂 to drag the column to another position.
  - Select to remove the column.
  - Select 🗷 to close the tab.
  - type: the column type. Depending on the type you select, the sort and search options will differ. Options:
    - text: if you select this type, the column content will be interpreted as text.
    - number: if you select this type, the column content will be interpreted as a number.
    - date (ISO-format): if you select this type, the column content will be interpreted as a date.
  - Title: the title of the column that is displayed on the column header.
  - Key (path): the key of the database collection that you want to display in the column. The key should correspond to the key of the text fields in the form you want to fill. The key can also be an array.
  - Sortable: if you select this checkbox, you can sort the content in the column.
  - Searchable: if you select this checkbox, a search field appears and you can search the content in the column.
  - Width: the column width.
- Maximum Rows: the maximum number of rows in your table.
- **only selectable**: if you select this checkbox, you will only be able to select an item from a table. It will not be possible to enter a value in the field. Select this option to prevent that a value is entered that does not exist.

#### Example

You have a form with the following text fields:

Name		≣Q
ID	type here	
Info	type here	
Country	type here	
Contact	type here	

You also have a **customobjects** database collection that contains the collection **customers** (customobjects.customers). This collection contains a record for each customer, containing the fields **\_id**, **name**, **info** and **country** ( for all of which the dot notation is **customobjects.customers.X**. For example, customobjects.customers.name). The content of customobjects.customers is this:

```
[
{
    "_id" : ObjectId("5a676619c60c00ad0000005"),
    "name" : "Client A",
    "info" : "blocked",
    "country" : "NL"
},
{
    "_id" : ObjectId("5a6766f3ea0c00e30000005"),
    "name" : "Client B",
    "info" : "betaclient",
    "country" : "BE"
},
{
```

You can use this information in the database to automatically fill in the values of the text fields. To do this, configure the KIOSK parameters the following way:

- Text field Name:
  - Type: Auto fill by table
  - (variable) Key: name
  - Label: Name
  - Collection: Custom objects
  - Custom object name: customers
  - Columns:

Column that will retrieve database information for Name text field	Column that will retrieve database information for ID text field	Column that will retrieve database information for Country text field	Column that will retrieve database information for Info text field	Column that will retrieve database information for Contact text field
Type: text	Type: text	Type: text	Type: text	Type: text
Title: name	Title: ID	Title: country	Title: info	Title: contact
Key (path): name <sup>(*)</sup>	Key (path): _id <sup>(*)</sup>	Key (path): country <sup>(*)</sup>	Key (path): info <sup>(*)</sup>	Key (path): contact <sup>(*)</sup>
Searchable: yes				

**Note:** <sup>(\*)</sup> this key needs to be identical to the key of the corresponding text field.

- Text field ID:
  - Type: Text
  - (variable) Key: \_id
  - Label: ID
- Text field Info:
  - Type: Text
  - (variable) Key: info
  - Label: Info
- Text field Country:
  - Type: Text
  - (variable) Key: country
  - Label: Country
- Text field Contact:
  - Type: Text
  - (variable) Key: contact
  - Label: Contact

Select one item by double click X			
name	ID	country	info
Client A	5a676619c60c00ad00000005	NL	blocked
Client B	5a6766f3ea0c00e300000005	BE	betaclient
Client C	5aa119348761655a34d73a53	BE	
Client D	5aa129a98761655a34d73a57	BE	blocked
	٠		

As a result, the Name field in the form contains a search button. If you select the search button, the columns appear with the values in the database:

Note that there is no **contact** column. This is because there is no object in the database with a key that corresponds with the key in the column (**contact**).

If you double click a row, the values of the database corresponding to that row are injected in the form. In this example, double clicking the row of the object corresponding to **Client B**, the available values are injected in the form:

Name	Client B	≣Q
ID	5a6766f3ea0c00e300000005	
Info	betaclient	
Country	BE	
Contact	type here	

The fields that are changed are displayed with a yellow background color. The field **Contact** does not receive any value, since the database does not contain an object with the key that corresponds to the key of the text field (**contact**). The field cannot retrieve information from the database, so it needs to be filled in manually.

## Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

## Form Actions

With Form Actions you can show or hide a field, depending on the value of another specified field.

When

• Is equal to: here you can specify a value. If the value that is entered in the active field in the form is equal to the value that is specified here, another field is shown or hidden.

### Then

• Field: here you can define the field that must be shown or hidden when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Select **b** to select the field you want to show or hide.

• Action: here you can define the action that will be performed when the value that is entered in the form is equal to the value that is specified in the parameter Is equal to. Possible actions are show or hide.

## Simple example

You have a form with two text fields:

- Label 1
- Label 2

Label 1 has a configured Form Action:

#### When

• Is equal to hide

#### Then

- Field id2
- Action hide

The result is that Label 2 will be hidden in the form when the value of Label 1 is equal to hidden.

#### Hidden

With **Hidden** you can hide a parameter field in the form. This is useful for example when you want to pass data that you do not want the users to see.

- **Type**: the type of the parameter.
- (variable) key: the name of the variable that will contain the parameter value. The variable will be addressed in the workflow with this name. You can use dot notation if you want to group single variables into a single object. For example: customer.name, customer.address and customer.telephone will result in one variable customer that is an object with three fields name, address and telephone.
- Help text: the help text that will appear next to the label.
- Field options:
  - **Required?**: if you select this checkbox, the field must be filled in. This will be indicated by an asterisk next to the label in **KIOSK**.
- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### Title

With **Title** you can create a title.

• **Type**: the type of the parameter.

#### Value Options

- **Text**: the type of the parameter.
- Type of title: the type of title (Title, Subtitle or Text).
- Text align: the alignment of the title (Left, Center or Right).

## Style

- **Background color**: the background color of the field.
- Font color: the font color of the text in the field.

#### Image

With Image you can add an image.

• **Type**: the type of the parameter.

## Value Options

• URL of image: the URL of the image.

- Width (px): the width of the image in pixels.
- Height (px): the height of the image in pixels.
- Horizontal align: the horizontal alignment of the image (left, right, center).

#### Style

- Background color: the background color of the field.
- Font color: the font color of the text in the field.

#### **Error Message**

With Error message you can add a placeholder for an error message.

- **Type**: the type of the parameter.
- Background color: the background color of the field.
- Font color: the font color of the text in the field.
- **Note:** The error message field is a placeholder. You cannot change the content of the error message.

# Users and Contacts

In CLOUDFLOW, there is a difference between Users and Contacts.

#### Users

Users can interact with the CLOUDFLOW system. They have access to the CLOUDFLOW Workspace and depending on their permissions they can manage specific parts of the CLOUDFLOW system. In CLOUDFLOW there is a special **admin** user who has all the permissions to manage the CLOUDFLOW system. Users have an **email**, a **login**, a **password** and **permissions**.

#### Contacts

Contacts are special guest users who cannot log into the CLOUDFLOW Workspace. The purpose of a contact is to view assets without the need of creating a user in CLOUDFLOW. They interact with CLOUDFLOW in a limited security context. They can be created manually in CLOUDFLOW Workspace, or are created automatically when needed. Contacts have only an **email**.

## Manage contacts

In MANAGE CONTACTS you can add new contacts to the contact list and edit existing ones.

#### **Contacts overview**

The contacts list provides an overview of the existing contacts. Contacts have multiple columns:

- NAME: the full name of the contact.
- EMAIL: the e-mail address of the contact.
- **ATTRIBUTES**: the attributes of the contact.
- SCOPE: the scope that is assigned to the contact.
- LAST LOGIN: the last time the contact has logged in.

#### Add contacts

Contacts can be added manually or automatically.

#### Manually

To add a contact manually, follow these steps:

1. Open the USERS tab.

- 2. Open the MANAGE CONTACTS tab.
- 3. Select Add Contact.
- 4. Define a name for the contact.
- **5.** Define the email address.
- 6. If needed, define a scope for the contact. Options:
  - **a.** All Scopes: if you select this option, the contact has access to all scopes. This means that the contact doesn't have any restrictions for accessing files.
  - b. Any other scope: if you select this option, the user only has access to the files of the selected scope.
- 7. If needed, add one or more attributes. Attributes can be used to identify a participant in an approval flow. See Attributes in Start Approval node for more info. To add an attribute:
  - **a.** Enter an attribute name.
  - **b.** Select +.
- 8. Select Save.

#### Automatically

In some cases, contacts are added automatically to the contacts list.

#### Example

If you invite a user that is not known to the system to view a file (see Invite Users To PROOFSCOPE for more information), the email address is added automatically as a contact.

## Edit contacts

When the contact is added to the list, you can edit it according to your needs. To do so, hover with your mouse over the contact and select **Edit**.

## Filter contacts

You can filter existing contacts by NAME, EMAIL and SCOPE. To do so, enter a query in one of the column headers.

## Sort contacts

You can sort existing contacts by NAME, EMAIL, SCOPE and LAST LOGIN. To sort contacts, select the corresponding column header.

## Manage users

In MANAGE USERS you can add new users to the users list and edit or delete existing ones.

#### **Users** overview

The contacts list provides an overview of the existing contacts. Contacts have multiple columns:

- USERNAME: the user name of the user.
- **PASSWORD**: the password of the user.
- NAME: the full name of the user.
- EMAIL: the e-mail address of the user.
- ATTRIBUTES: the attributes of the user.
- **SCOPE**: the scope that is assigned to the user.
- LAST LOGIN: the last time the user has logged in.

## Add users

1. Open the USERS tab.

- 2. Open the MANAGE USERS tab.
- 3. Select Add User.

=

=

- 4. Define a USERNAME.
- 5. Define a PASSWORD.

**Note:** You can add password rules in the script of the system flow called **User Management Flow**. See **System flows** for more information on system flows.

- **6.** Define a name for the user.
- 7. Define the email address.

**Note:** A user's email address needs to be unique. A recurring email address related to different users will result in conflicting behaviour.

- 8. If needed, define a scope for the user. Options:
  - **a.** All Scopes: if you select this option, the user has access to all scopes. This means that the user doesn't have any restrictions for accessing files.

**b.** Any other scope: if you select this option, the user only has access to the files of the selected scope.

9. Define the **permissions**. Depending on these permissions, the user can of cannot perform specific actions.

Permission	Functionality	
Admin user	The user has administrator rights. See Admin user and administrators on page 285 for more information on the difference between the admin user and administrators.	
Patchplanner	The user has access to <b>PATCHPLANNER</b> .	
Manage Chains	The user can access APPROVAL > MANAGE APPROVAL CHAINS. (Deprecated)	
Manage Users	The user can manage users.	
Manage Scopes	The user can manage scopes.	
Manage Templates	The user can access <b>APPROVAL</b> > <b>MANAGE</b> <b>APPROVAL TEMPLATES</b> .	
Manage Whitepapers	The user can manage workflows. The user has access to the workflow editor via a link in <b>KIOSK</b> in case the jacket errors. The user can also manage the Jobs in the Job list in KIOSK and use the rules <b>All</b> or <b>Just Me</b> . If the user selects <b>All</b> , the user can manage all Jobs from other users.	
Manage Share	The user has access to <b>SHARE</b> and can set up <b>SHARE</b> networks.	
Manage Assets	The user can manage assets in <b>ASSETS</b> (download, rename, delete, cut, copy, paste).	
External user	The user is an external user who is logged in on a custom page in CLOUDFLOW and is prevented from (accidentally) navigating to the standard CLOUDFLOW pages, with the exception of <b>PROOFSCOPE</b> viewer.	
Manage Jobs	The user can manage Jobs.	
May Upload	The user can upload or download a file to/from a file store in <b>ASSETS</b> .	

Permission	Functionality	
May Delete Jackets In Kiosk	The user can delete jackets in <b>KIOSK</b> with <b>t</b> .	

**Note:** There is a difference between scopes and permissions. See Scopes versus permissions on page 286 for more information.

10. Select Save.

=

- **11.** If needed, define one or more attributes. Attributes can be used to identify a participant in an approval flow. See Attributes in Start Approval node for more information.
- **Note:** Each user you create also appears in the contacts list. If you want to add an attribute, you need to add it in the **CONTACTS** tab. See Add attributes for more information.

#### Edit users

To edit a user, hover with your mouse over the contact and select Edit.

=

**Note:** When you create a new user or change an existing user's password, an email is sent to the user's email address. The email contains the username and password. You can change the template mail in Approval templates on page 301.

## **Delete users**

To delete a user, hover with your mouse over the contact and select Delete.

#### Filter users

You can filter existing users by USERNAME, NAME, EMAIL and SCOPE. To do so, enter a query in one of the column headers.

## Sort users

You can sort existing users by USERNAME, NAME, EMAIL, SCOPE and LAST LOGIN. To sort users, select the corresponding column header.

## **User Preferences**

Users can edit user settings via the User Preferences.

To open the User Preferences, select the User in the top right of the window and select PREFERENCES.

User preferences overrule the corresponding default settings defined in **SETTINGS** on page 47. This way, users can change their:

- UI language
- Length unit
- Small length unit
- Distortion unit
- Text unit
- Resolution unit
- Ruling unit
- Password

# Admin user and administrators

In CLOUDFLOW there is an Admin user and there are one or more administrators.

What is the difference?

#### Admin user

The admin user is a superuser.

- The admin user has all the rights and privileges that exist in the system.
- The admin user is a part of the system by default.
- You cannot delete the admin user.
- You cannot edit the admin user's username or password.

## User with administrator rights

Administrator are users who have administrator rights.

- You can grant a user administrator rights by selecting the checkbox Admin user in USERS > MANAGE USERS.
- You can delete or edit administrators.
- Administrators have the same rights as the admin user.
- Administrators automatically have all user permissions, except the **External user** permission. See User permissions for more information about user permissions.

# **Scopes**

For access and security reasons, CLOUDFLOW uses the concept of scopes.

A scope is like a glass bell with objects (files, assets, users, workflows...) underneath. The glass bell or scope is a method to restrict the access to certain objects.

A scope is assigned to a user or a contact.

- One scope can be assigned to several users.
- A user or contact can be assigned with maximum one scope.
- If a user or contact is not assigned with a scope, this means that this user or contact has access to all scopes. This is the default.

When a user is assigned with a scope, the user can only access the objects in that scope.

## Scopes versus permissions

Scopes and permissions are two different concepts:

- Scopes restrict a user's access to certain objects.
- Permissions define if a user can perform certain actions. See Manage users on page 283 for more information on permissions.

## Example

If you want a certain user to be able to delete Jackets in KIOSK, select the permission checkbox **May Delete Jackets In Kiosk** in the **MANAGE USERS** tab.

However, if you don't want the user to upload certain files, you need to put these files in a scope and make sure to **not** assign the scope to the user.

## Create a scope

To create a scope, follow these steps:

- 1. Open the USERS tab.
- 2. Open the MANAGE SCOPES tab.
- 3. Select Add Scope.
- 4. Define a name for the scope.

- 5. Create a filter that will define which files are part of the scope.<sup>5</sup>
  - The filter is built with a Regular expressions on page 475. For example, the filter **^PP\_FILE\_STORE** will give you access to all files in PP\_FILE\_STORE.
  - =

Note: You can use https://regex101.com as an online regex tester.

• The file path that you need to use for building the regular expression is different from the CLOUDFLOW URL.

Differences:

CLOUDFLOW URL	File path for filtering
Uses %20 for a space	Uses a regular space
Starts with <b>cloudflow:</b> //	Starts with the name of the File store
Example	Example
cloudflow://Fileserver3/01%20WIP/ files/FD000038/Olive_Oil_Z34EN.pdf	Fileserver3/01 WIP/files/FD000038/ Olive_Oil_Z34EN.pdf

**6.** Define a welcome page. <sup>6</sup>

The welcome page is an HTML file that will be visible for the user with a specific scope after logging in.

**Note:** Make sure the HTML file is in a scope that is accessible for the user.

#### 7. Select Save.

To edit or delete an existing scope, select the scope and select Edit or Delete.

You can filter scopes by NAME, FILTER and WELCOME PAGE. To do so, enter a query in one of the headers.

## Assign a scope

When you have created a scope, you can assign it to a contact or a user.

To assign a scope to a user follow these steps:

- Open the USERS tab.
- Open the MANAGE USERS tab.
- Hover with your mouse to select the contact and select **Edit** on the right side.
- Select the scope. Options:
  - 1. All Scopes: if you select this option, the contact has access to all scopes. This means that the contact doesn't have any restrictions for accessing files.
  - 2. Any other scope: if you select this option, the user only has access to the files of the selected scope.
  - **Note:** It is not possible to assign a scope to the Administrator.

You can assign a scope to a contact in a similar way by opening the MANAGE CONTACTS tab.

## **Guest scope**

=

A special note on the **Guest scope**.

The Guest scope is present by default. This scope is assigned to anyone who is not (yet) logged in to CLOUDFLOW. You cannot delete it, you can only edit it.

<sup>6</sup> This step is not compulsory. You can create a scope without defining a welcome page. In this case, the scope will only contain file access restrictions.

<sup>&</sup>lt;sup>5</sup> This step is not compulsory. You can create a scope without defining a filter. In this case, the scope will only contain a welcome page.

When a user or contact logs in and is assigned with a scope, this user or contact has access to:

- All objects in the scope the user or contact is assigned with.
- All objects in the Guest scope.

## Restricting access to workflows using scopes

You can restrict the access to workflows in the Start From Kiosk and Start From Web Request nodes.

In the field **Allow Access to Scopes** you can enter one or more scopes (separated by a comma). Consequently, only the users with these scopes assigned can access the input of the workflow.

## Scoping workables

You can define the scope of the workable in the Start from Kiosk and Start From Web Request nodes.

If you want that the access of the workable is restricted to the scope of the user that creates the workable, you need to enable the checkbox **Scope From User**.

If you don't want to restrict the access of the workable, you need to disable the checkbox.

#### Example

User A is assigned with a scope and therefore only has access to the files in File store X. User A starts a workable in a flow where a file will be copied to folder F which is not in File store X.

To make sure that the workable created by user A can reach Folder F, you need to disable the checkbox **Scope From** User. In case you enable the checkbox, the access of the workable is restricted to File store X, since this is the scope of the user A.

## How to filter assets with scopes

In some cases it is useful or needed to filter assets for one or more users.

For example, when a user is only allowed to see PDF files in a specific File store. In that case, you need to create a scope containing a (regular expression) filter and assign the scope to that user.

Here you can find some examples on how to filter assets by using scopes:

Scope	(Regular Expression) Filter	Example
Single Filestore - All Folders - All Subfolders - All Files	^FILESTORENAME	^PP_FILE_STORE
Single Filestore - All Folders - All Subfolders - only <b>.ext</b> Files	^FILESTORENAME/.*.ext	^PP_FILE_STORE/.*.pdf
Multiple Filestores - All Folders - All Subfolders - All Files / only <b>.ext</b> Files	^(FILESTORENAME_1  FILESTORENAME_2  FILESTORENAME_3 ) ^(FILESTORENAME_1  FILESTORENAME_2  FILESTORENAME_3 )/.*.ext	^(PP_FILE_STORE CMKCORE  CNHCORE)
Single Filestore - Single Folders - All Subfolders - All Files / only .ext Files	^FILESTORENAME/ FOLDERNAME ^FILESTORENAME/ FOLDERNAME/.*.ext	^PP_FILE_STORE/Marks
Scope	(Regular Expression) Filter	Example
---	---	--
Single Filestore - Multiple Folders - All Subfolders - All Files / only <b>.ext</b> Files	^FILESTORENAME/ (FOLDERNAME_1  FOLDERNAME_2  FOLDERNAME_3 ) ^FILESTORENAME/ (FOLDERNAME_1  FOLDERNAME_2  FOLDERNAME_3 )/.*.ext	^PP_FILE_STORE/(Marks HTML  XML)
Single Filestores - Single Folder - Multiple Folders - All Subfolders - All Files) / only <b>.ext</b> Files	^FILESTORENAME/ FOLDERNAME/ (FOLDERNAME_1  FOLDERNAME_2  FOLDERNAME_3 ) ^FILESTORENAME/ FOLDERNAME/ (FOLDERNAME_1  FOLDERNAME_2  FOLDERNAME_3 )/.*.ext	^PP_FILE_STORE/Marks(Single  SR Info)
Single Filestores - Single Folder - Any folder - 1 Unique Folder - Any Folder - All Files) / only <b>.ext</b> Files	^FILESTORENAME/ FOLDERNAME/.*/ UNIQUEFOLDERNAME/.* ^FILESTORENAME/ FOLDERNAME/.*/ UNIQUEFOLDERNAME/.*.ext	^PP_FILE_STORE/Marks/.*/ internal/.*

# Variables

To make workflows more dynamic, CLOUDFLOW uses the concept of variables.

Variables allow you to use values that depend on a certain context in the flow (for example retrieved from the filename or from an external system) and to set up your workflow in a flexible way.

# Example

In this case all incoming files in the workflow node are copied in a folder which is created on the fly if necessary. The path where the files will be copied to has two variables: **Folder Path** and **Filename**.

Copy File	?	×
File to convi		
Пе ю сору.		
All files from node previous Node	1	
To file or folder:		
Folder Path copy/ Filename	1	
Create folders		
Overwrite existing file		
Unique name mode: None		

There are two kinds of variables:

- Default variables, which are available by default in the system.
- Custom variables, which you need to define.

# **Defining custom variables**

There are various ways to introduce custom variables in a CLOUDFLOW workflow.

### Via Form Builder

You can set custom variables via the Form Builder on page 260.

When you enter a value in a form field, a variable is created. This can be used throughout the complete workflow. This is one of the ways variables are introduced in your flow and can be used throughout the flow.

### In the Set Variable node

You can set custom variables in the Set Variable node.

### In various other nodes

Various nodes contain the field Variable name:.

This field will create a variable containing the result of the node.

For example, the Get Meta Data on page 164 node, the **meta\_data** variable will contain the metadata of the incoming file.

### Via a script

You can set custom variables via the Script node.

### Via an API call

When you submit to a workflow using the public **API** you can introduce a list of custom variables with matching values to the workflow as an object.

# Example

```
In this example the variable email is introduced with value test@hybridsoftware.com to the workflow Demoflow:
api.hub.start_from_whitepaper_with_variables("Demoflow", "Input",
{"email":"test@hybridsoftware.com"})
```

## Via Expression Builder

You can introduce custom variables in Expression Builder.

# Variable specifications

There are some limitations you have to take into account when creating variables.

# Name

Variable names cannot contain the following characters:

- \$ (dollar)
- . (dot)
- / (slash)
- (backslash)
- # (crosshatch)

# Value

The value of a variable can be any valid JSON object. A JSON object is valid if the keys:

- Do not start with \$ (dollar).
- Do not contain dots.

# Encoding

Everything needs to be UTF-8 encoded.

# **Expression Builder**

With Expression Builder you can build for example a path by using expressions that contain text and/or variables.

# **Open Expression Builder**

In many nodes, there is a pencil icon on the right of the field. If you select it, Expression Builder is opened.

Copy File	?	×
File to copy: All files from node previous Node		
To file or folder:		
Create folders		
Overwrite existing file		
Unique name mode: None		
	_	

**Note:** In some cases the pencil is not visible immediately. In that case you can use **ALT-Click** in the field to visualize it.

The following window appears:

=

Start Approva	I / references				×
Q Search All File Info Misc General Workable Info Job	Filename Filename w/o ext. Filename ext. Full file url Folder Path	Expression 🕢 Value type: As is 🕞	Add Path	Original S	itate
Date/Time Custom Variables Add Variable Filename The name of the first	t output file of a node.	Variables Folder Path: cloudflow://PP_FILE_STORE/inputfiles/ Result cloudflow://PP_FILE_STORE/inputfiles/			
			Cancel	Sa	ve

# Create an expression

To build an expression, follow these steps:

- 1. Add the elements you want to use in the expression. You can use:
  - Variables. To add a variable, follow these steps:
    - **a.** In the left column (**A**), you can see the list of variables, divided into variable categories. Select a variable you want to add to your expression. If needed you can use the filter on top. In **B** you can see some information about the variable you selected.
    - **b.** Drag the variable to the area where you can build the expression (C).
  - Hard text. To add hard text, type the text in the area where you can build the expression.
  - A hard path. Select Add Path on the top right to browse your CLOUDFLOW File stores and add a hard path.
- 2. If needed for testing, edit the output of the variables in D.
- 3. View the result of the expression you have built in E and tweak it if needed.
- 4. Select Save.

Start Approval / references		
Q Search All Filename Filename w/o ext.	Expression  Value type: As is Folder Path	Add Pa
Filename ext. Misc Full file url General Folder Path	С	
Workable Info Job Date/Time		
Custom Variables Add Variable	Variables Folder Path: cloudflow://PP_FILE_STORE/inputfiles/	
Filename The name of the first output file of a node.	Result cloudflow://PP_FILE_STORE/inputfiles/	
B	<b>E</b>	
		Ca

# **Custom variables**

**Custom Variables** are variables that are not by default present in the system and that were introduced by a custom action.

See Defining custom variables on page 290 for more information on how to introduce variables in your workflow.

If you use a custom variable in your expression but the variable itself is not present in the workflow, the flow will error. To avoid this, you can define a default value for a custom variable in case the variable does not exist. To do this, follow these steps:

- 1. Select the custom variable.
- 2. Select the checkbox next to Use default value if variable does not exist.
- 3. Define the value in the field **Default Value**.

### Adding a variable

If a specific variable is not available, you can create it by adding a variable.

In **F** you can add a variable. The purpose of this option is to be able to build an expression using variables that have not yet been introduced in the current node of the workflow, but that will be introduced later in the workflow. This way it is not necessary to first run the *workable* until the node where the variable is defined to be able to use it. However, if you run the workflow, the variables will become available in Expression Builder without adding them manually.

Follow these steps:

- 1. Select Add Variable.
- 2. Enter the variable you want to add in the field.
- 3. Select OK.

The variable will be added to the category Custom Variables

Start Approval / references		
Start Approval / references         Q Search         All         File Info         Misc         General         Workable Info         Job         Date/Time         Custom Variables         Add Variable         Filename         Filename w/o ext.         Full file url         Folder Path	Expression Value type: As is C Folder Path	Add Pa
		Ca

# Variable options

Ę

You can select a default variable and edit the options to change the output of the variable.

Note: Adding variables here is not one of the ways to introduce new variables.

You can select the variable in the parameter field in the node or in the Expression builder editing window.

Filename from node: Previous
Notation: URL encoded O

The options that you can edit depend on which variable you have selected. Some examples:

- In case of a **Filename**, you can select another node to use the Filename variable from and you can edit the notation.
- In case of a Job name, you can define a dot notation to edit the path.
- In case of a Year, you can edit the date format.



Note: When the node changes, you need to reconfigure the variable.

## Modifiers

You can select a variable in your expression and modify it to customize the result of the variable value, based on the position of its characters or on its length.

To add a modifier based on the character position, follow these steps:

- 1. Select the variable (you can do this in the parameter field or in the Expression builder editing window).
- 2. If needed, edit the options of the variable.
- **3.** Select one of the two options:
  - Characters Of in case you want to modify the variable based on the characters.
  - Length Of in case you want to modify the variable based on the length.

### Example 1: based on the characters.

There is a variable **Cusname** in your workflow and you want to use it to build a path. However, you don't want to use the full result of the variable, but only the first three characters of the variable value. To do so, select the **Cusname** variable and select **Add Modifier**. Configure the modifier like this:

Filename	×
Filename from node:     Previous       Notation:     URL encoded	
Characters Of	×
starting from:       the character at position       1       from the beginning       Image: Constraint of the character at position       Image: Constrainton at position </th <th></th>	
Characters Of 🗘 Cancel S	ave

This will result in an expression that contains the first three characters of the customer name in stead of the complete customer name.

By selecting **Add Modifier**, you can define more than one modifier for a variable. Note that in that case, the system will read the modifiers top-down, where the second modifier is extracted from the outcome of the first modifier. Select **Save** to save your changes, select **Cancel** to discard your changes.

In the example below, the first modifier will read all characters starting from the first underscore till the end of the customer name value. The second modifier will start from that result and will therefore build a value containing the first three characters starting from the underscore.

cusname	×
Variable: cusname	
Default Value:	
Characters Of	×
starting from:       the 1st occurrence of         from the beginning          to:       the character at position       1       from the beginning	
Characters Of	ж
starting from:       the character at position       1       from the beginning       2         to:       the character at position       2       3       from the beginning       2	
Characters Of O Cancel Sa	ve

You can also use the length of the selected node outcome.

- In case of a string this is the number of characters.
- In case of an object and array this is the number of entries.
- In case the other objects this is 0 or 1.

# Example 2: based on the length

Before submitting a job, the user needs to enter all the customers on a customized form, where all the customer names are stored in the variable key **Clients**:

0			
		Customers	C

INPUT Input Route OPTIONS Customers Customer A Customer B Customer				ŕ
Input Route OPTIONS Customers Customer A Customer B Cus	INPUT			
OPTIONS   Customers   Customer B     FILES   Upload   Cloudflow   Browse files	Input	Route		\$
Customer A   Customer B   FILES   Upload   Cloudflow   Browse files	OPTIONS			
Customer B   FILES   Upload   Cloudflow     Browse files	Cus	tomers	Customer A	0 0
FILES Upload Cloudflow Browse files			Customer B	00
FILES Upload Cloudflow Browse files				00
Upload Cloudflow				
Browse files	Unload	Olavalfia		
Browse files	Upload	Cioudilo	w	
Browse files				
Browse files	]			
Browse files				
	Browse file	es		
Submit			Submit	

You want to route on the number of customers the user enters in the form: 1 when the user enters 1, 2 when the user enters 2, 3 when the user enters 3 and mismatch for all the rest of the entries:



### In this case you can use the length of the variable key that you used in the customized form (Clients):

oute		
Value:		1
Value type: Number 🗢		
Routing: Compare to		
2		
3	3	
Routing Values type: Number		
Compare Numbers Accuracy: 0,001		
Case Free String Compare:		
		Close

The list is an array, so if the user enters 1 customer, the length is 1 and the workable will route to output 1.

### Change the syntax

### Change the syntax

In G you can change the syntax to XML, JSON or Text.

- With this functionality you can enter XML, JSON or Text content and if needed, combine it with variables.
- The system will warn you in case of syntax errors with a warning or error symbol. If you hover over it, more information will be given.
- If you select Expression, the default expression window will open again.

# Change the value type

### Change the value type

In **H** you can change the value type. Options:

- As Is: if you select this option, the expression will be interpreted as is.
- Text (String): if you select this option, the expression will be interpreted as a text sting.
- Number: if you select this option, the expression will be interpreted as a number.
- True/False (Boolean): if you select this option, the expression will be interpreted as a True or False value.

# Approvals

With CLOUDFLOW Workspace you can build approval cycles and have participants approve or reject a file.

# Build an approval cycle

You can build an approval cycle in the workflow area.

# Steps

The **minimum steps** you need to take are:

- 1. Create a Start Approval node: this will start the approval cycle.
- 2. Create a Wait For Approval node: this node will wait for the individual approvals and puts the workable on hold. The workable will be released when the file is approved, rejected or delegated. See Approve or reject a file on page 300 for more information.
- **3.** Create a Route After Approval node: this node will collect all the individual approvals, and send the *workable* to a certain flow-path depending on the assessments of all participants.
- 4. Connect the nodes this way:
  - a. Connect the success output of the Start Approval node to the input of the Route After Approval node.
  - b. Connect the approvals output of the Start Approval node to the input of the Wait for Approval node.
  - c. Connect the success output of the Wait for Approval node to the approvals input of the Route After Approval node.

For complete flexibility, you can add many more steps to your approval cycle :

- Add a Prepare PROOFSCOPE node: this will prepare the necessary data to view the rendered file in PROOFSCOPE.
- Add a Generate Proofscope URL node: this will prepare a URL that can be used to view one or more files in PROOFSCOPE.
- Add a Send E-mail node: this will send invitations by e-mail to all participants in the approval.

• ...

# **Multiple participants**

You can add multiple participants to the approval cycle by selecting the + icon in the Start Approval node.

A separate workable will be created for each participant.

All workables are routed to the **Route After Approval** node. This node waits until all participants have made an approval decision and collects these decisions. Depending on the **Policy** in the node, the file receives an *overall* status and is approved or rejected. Policy options:

- All Participants need to Accept: all participants need to approve the file.
  - If all participants approve, the file is approved.
  - As soon as one of the participants rejects, the file is rejected.
- All Participants need to assess: all participants need to either approve or reject.
  - If all participants approve, the file is approved.
- If one of the participants rejects, the file is rejected, but only after all participants have assessed the file.
- Only One Participant needs to Accept or Reject: the first participant determines the routing.
  - As soon as one of the participants approves, the file is approved.
  - As soon as one of the participants rejects, the file is rejected.

# **Combined approval**

A combined approval is an approval where several approval steps are involved.

- Steps can be added
  - by selecting the Combined checkbox in the Start Approval node, or
  - by selecting Add Step in the APPROVAL tab.

The final status depends on the combination of the outcome of the approval steps.

# Manage approvals

How to manage approvals.

### Approve or reject a file

You need to approve or reject a file to release the workable.

In an approval cycle, the workable is automatically put on hold in the **Wait For Approval** node and the asset will have the status **Waiting Approval**.

To release the workable, you need to approve or reject the file. There are various ways to do this:

### In KIOSK

In KIOSK you can approve or reject a file, without opening it in PROOFSCOPE first. If you select one of the decision buttons, the workable will continue.

### In PROOFSCOPE

When the file is prepared to be viewed in PROOFSCOPE with the **Prepare PROOFSCOPE** node, you can approve or reject it in PROOFSCOPE. See Approve or reject a file in PROOFSCOPE for more information.

After a file has been approved or rejected, the status will change accordingly.

### Force pending approvals

Pending approvals are approvals that have not been accepted or rejected yet. In case of an approval with multiple participants, it is possible to approve or reject the file without having to wait for the participants' decisions.

You can force pending approvals in different ways.

### In HOME

The default **HOME** tab in CLOUDFLOW gives an overview of the pending approvals. The pending approvals are marked with a question mark.

If you select a pending approval, the asset details open. The asset is indicated as **Waiting approval**. Select **APPROVAL** to open the **APPROVAL** tab. Here you can select an action in the drop-down list:

- Force Approved: the asset will be approved.
- Force Rejected: the asset will be rejected.
- **Cancel**: the approval will be canceled.

### In KIOSK

KIOSK gives an overview of all the jackets. To force an approval in **KIOSK**, select the jacket and select **Force Approved**, **Force Rejected** or **Cancel**.

### In ASSETS

In ASSETS, you can filter on all assets with pending approval if you enter status:pending in the search window.

**Note:** To find all approved assets, enter **status:accept**. To find all rejected approvals, enter **status:reject**.

If you select a pending approval, the asset details open. The asset is indicated as **Waiting approval**. Select **APPROVAL** to open the **APPROVAL** tab. Here you can select an action in the drop-down list:

- Force Approved: the asset will be approved.
- Force Rejected: the asset will be rejected.
- Cancel: the approval will be canceled.

# In APPROVAL

In APPROVAL you have an overview of the following approvals:

- ALL APPROVALS: this tab gives an overview of all the approvals.
- MY APPROVALS: this tab gives an overview of your approvals.
- MY TEAM APPROVALS: this tab gives an overview of your team approvals. The team members include the Team Leaders and the Observers that you can define in the Start Approval node.

In all these tabs you can filter on the pending approvals by selecting **Pending** next to **ASSESSMENT**.

You can select an action in the drop-down list:

- Force Approved: the asset will be approved.
- Force Rejected: the asset will be rejected.
- Cancel: the approval will be canceled.

In these tabs you can also edit pending approvals. To do so:

- 1. Select the arrow next to the thumbnail of the asset for more details.
- 2. Select 🏟 to activate the edit mode. The following edits are possible:
  - Select I to change the email address of an existing participant. The assessment of the wrong email address will be delegated to the correct email address, so a new participant will be added.
  - Select **x** to cancel existing participants. The participants remain in the overview, but their assessment will be set to **Canceled**.
  - Select + to add new participants.

# **Approval templates**

You can build approval templates in both plain text and HTML to use them in a workflow or approval cycle.

To add an approval template, follow these steps:

- 1. Open APPROVAL > MANAGE APPROVAL TEMPLATES tab.
- 2. Select Add Template.
- **3.** Enter a name for the template.
- 4. Select a language.
- 5. Enter a subject. The subject can contain dynamic content.
- 6. Select Save.
- 7. Select the template.
- 8. Select Edit.
- 9. Enter or edit the email body. The email body can use dynamic content.
- **10.** Select **HTML** if you want to use HTML syntax.
- 11. Select Save.

# **Dynamic content**

Dynamic content is content that will be substituted during processing. The content is derived from information found in the workable, notes, approvals...

Dynamic content in email templates needs to start with \${ and end with } (for example: \${variable.var1}). During processing, the system will look for the information and replace the content between \${ and }.

You can use different data types of dynamic content:

# Variables

Keyword: variable

This data type will insert the value of a specified variable into the text. To select the actual variable you need to use a **dot notation**. For example, variable.var1 will select the variable **var1**.

You can also use sub elements of a variable. For example, if your variable **object** is {'element1': 'value1', 'element2': 'value2'}, you can select the first element by using variable.object.element1.

- If the selected variable is a simple string or value, the resulting value will also be a simple string or value.
- If the selected variable is an array or object, the complete object will first be stringified as a JSON object and then inserted.

### Examples

```
${variable.var1}
```

\${variable.var1.subvar}

A warning will be generated if the selected variable cannot be found in the workable.

### Variables Formatting options

• **PreEncoded**: in case you want to allow to insert data directly into the HTML stream and you want to make sure that the inserted data is correct HTML, you can use the **PreEncoded** option:

\${variable.var1 PreEncoded}

• IsPreflightData: output from the PACKZ Analyze node and from the PitStop Preflight node generate a well structured JSON variable. The IsPreflightData option allows you to include that data in an email and the results will be formatted into a table:

\${variable.var1 IsPreflightData}

### Workable Messages

### Keyword: workable\_messages

This data type will insert the messages that were generated during processing of a workable in the text.

- In case of a plain text email, the messages will be generated as a list of entries.
- In case of an HTML email, the messages will be placed in a table consisting of three columns.

#### Example

\${workable messages}

An empty string will be generated if there are no messages in the workable.

### **Approval Overview**

### Keyword: approval\_overview

This data type will insert an overview of the approval in the text. It will add the approval name and the assessment for each participant and/or sub-approval.

- In case of a plain text email, the overview will be generated as a sequence of the name and assessment, and then a list of participants and assessments.
- In case of an HTML email, the overview will be generated as a sequence of the name and assessment, and then either a table with the participants, or a table with sub approvals, with a nested table for the participants examples.

### Example

\${approval overview}



**Note:** The approval data that will be inserted is derived from the approval linked to the workable.

### **Notes Overview**

Keyword: notes\_overview

This data type will insert the notes into the text. It will add the user and the content of each comment.

- In case of a plain text email, the overview will generated as a list of users and comments.
- In case of an HTML email, the overview will be generated as a table with the users and comments.

### Example

\${notes overview}

# How to avoid issues when sending out approvals

These are recommendations to avoid problems when sending out approvals.

### Set the port of the external CLOUDFLOW link to 80

The external link to access CLOUDFLOW (for example: http://cloudflow.example.be) often contains a port (for example: http://cloudflow.example.be:80). Recommended is to use port 80, which is the default for http. If not, the link is often blocked when the CLOUDFLOW link is accessed externally.

### Make sure the firewall of the external party allows the CLOUDFLOW link

If the firewall level set by the external party is too high, it may prevent access to the CLOUDFLOW URL http:// cloudflow.example.be. In that case, make sure the firewall allows access to the CLOUDFLOW link.

### Make sure that the CLOUDFLOW server has sufficient upload speed

If an external party opens PROOFSCOPE to approve a file, the data are uploaded on the CLOUDFLOW server where CLOUDFLOW is installed to send these to the external party. However, if the CLOUDFLOW server does not have sufficient upload speed, this process will be slow.

### Make sure that the sender email is known by the mail server

A mail sent from CLOUDFLOW contains a sender email (for example sender@hybridsoftware.com). Make sure that this sender address is known by the mail server. If not, the sender address is blocked by the mail server.

### Add the sender email address as a trusted sender

In some cases, a mail sent from CLOUDFLOW is considered as a junk mail. You can solve this by adding the sender email address (for example sender@hybridsoftware.com) as a trusted sender in the mail client settings of the customer.

# PROOFSCOPE

PROOFSCOPE is an application for soft proofing and assessment.

With PROOFSCOPE you can do the following:

- View files
  - PDF, JPEG, BMP, CF2 files.
  - Differences between files.
  - 3D files.
  - 1 bit TIFF files and LEN files.
- Compare files

- Assess files
  - Approve
  - Reject
  - Delegate

# **View PDF files**

With PROOFSCOPE you can view PDF files.

To view a file in PROOFSCOPE, select the file from the ASSETS tab and select View File<sup>7</sup>.

PROOFSCOPE opens:



It contains the following parts:

- A: the Sidebar.
- **B**: the display area with the **Toolbox** on top.
- C: buttons to **download** the file or the report, or buttons to approve the file in case of an approval.

# Sidebar

On top of the left Sidebar, you can switch between the different view modes.

# Thumbnails

**Thumbnails** gives you an overview of the pages.

In case you're viewing a multipage file, all pages will be displayed.

<sup>7</sup> in case you want to view a TIFF file created by the CLOUDFLOW standalone RIP, you can open it from RIP > Jobs

<sup>&</sup>gt; Files. See Standalone RIP on page 349 and View screened files on page 310 for more information.

# Notes



**Notes** gives you an overview of the notes.

You can view notes per user or per page.

- Hide notes: if you select this button, all the notes on the design are hidden.
- Show notes: if you select this button, all notes on the design will be shown.
- Show notes of previous versions: if you select this checkbox, the notes from older versions of the file are shown.

# Reports

**Reports** gives you an overview of the preflight results in the **Analyze node**.

If you select a checkbox, the problem area of a specific type of preflight result is indicated by a red square. If you select the triangle, the list is expanded. If you select a specific preflight result, the error becomes green transparent. When zoomed in, the zoom will automatically shift to the error area.



Note: This view is only available if the checkbox Show Preflight Notes in the Generate proofscope URL node is selected.

# Separations

**B** Separations gives you more information about the file.

- SEPARATIONS: here you can view and edit the separations.
  - Select or deselect the checkboxes next to the separations names to display or hide separations.
  - Select Edit... to activate the editing mode. When editing mode is active, a pencil is displayed in the separation color box. You can perform the following actions:
    - Change the separation name. To do this, select the pencil to open the separation parameter box. Select the name to change it and select Save.
    - Change the spot colors. To do this, select the pencil to open the separation parameter box. Change the color by moving the color picker or entering a different spot color name. Select Save.
    - Change the opacity of the spot colors. To do this, select the pencil to open the separation parameter box. Select or deselect the **opaque** checkbox.
    - Change the separation order by dragging and dropping the separations to a different order.

Select Edit... again to deactivate the editing mode.

# SEPARATION OPTIONS

- Single separation as black: if you select this checkbox, the single separation is displayed in black. This option • only makes sense if you have selected one separation in the SEPARATIONS list.
- Invert separation order: if you select this checkbox, the separation order is inverted.

**Note:** This option is only for previewing purposes and does not change the actual order of the separations.

- **INSPECTION:** 
  - Show inspection overlay: if you select this option, the design is checked against specific characteristics: •
    - Minimum dot: if you select this checkbox, a red overlay layer will highlight the areas in the design that • have a lower ink percentage than the minimum percentage predefined in the First visible dot field.
    - Maximum dot: if you select this checkbox, a red overlay layer will highlight the areas in the design that have a higher ink percentage than the maximum percentage predefined in the Maximum dot field.
    - Total area coverage: if you select this checkbox, a red overlay layer will highlight the areas in the design that have a higher total area coverage than the percentage defined in the Coverage field.

- **PAGE BOXES**: here you can select which page boxes are displayed. Select **Show more...** for more information about the page boxes, such as the width and height, and the bleed distance in case of a bleed box.
- PAGE SIZE: here you can view information about the page size of the design.

# **Approval Form**

Approval Form displays the approval form that is created in the Start Approval node. In case one or more fields are required to be filled in, the user cannot approve or reject the file as long as these required fields are empty. The Approval Form is only visible when the checkbox Display Form in the Start Approval node is selected.

# Toolbox

The Toolbox on top allows you to switch between different types of view tools.

Icon	Description
:	<b>Show Sidebar</b> allows you to open or close the <b>Sidebar</b> on the left.
•	<b>Zoom in</b> allows you to zoom in. If you selected the <b>Thumbnails</b> view in the <b>Sidebar</b> , the part on which you have zoomed is highlighted.
100%	<b>Zoom</b> allows you to enter a zoom percentage.
0	<b>Zoom out</b> allows you to zoom out.
8	Fit allows you to fit the complete design in the window in case you are zoomed in or out.
5 C	<b>Rotate</b> allows you to rotate the design to the left or to the right.
<b>य</b> म	Mirror allows you to mirror the design.
Q	<b>Zoom</b> allows you to magnify a selected part of the file. If you click, you zoom in, if you ALT + click, you zoom out.
<del>4</del>	<b>Pan</b> allows you to use panning on the design while zoomed in. To pan, hold down the left mouse button and move the mouse. If you selected the <b>Thumbnails</b> view in the <b>Sidebar</b> , the part on which you have zoomed is highlighted. You can move this highlighted part around by holding down the left mouse button and moving the mouse to change the panning zone.

Icon	Description	
	<b>Notes</b> allows you to add notes to the design. You can add several types of notes:	
	<ul> <li>Note will create a note on the location where you click in the design.</li> <li>Bestangle allows you to draw a restangle.</li> </ul>	
	around the area of the design you want to draw attention to.	
	<ul> <li>Oval allows you to draw an oval around the area of the design you want to draw attention to.</li> <li>Freehand allows you to draw a free form</li> </ul>	
	around the area of the design you want to draw attention to.	
	around existing live text in the design. The selected text will then be copied to the text box.	
	Note: The note comments are automatically saved after 5 seconds sec of inactivity or when the note dialog is closed.	
	Note: Saving text in a text box before editing it for comparison purposes can be useful in an approval flow.	
Ð	Note: To use this type of note, make sure that Text Layer Support is enabled in the PROOFSCOPE SETTINGS. In case of an approval workflow, make sure the checkbox Show Note-From-Text Tool is enabled in the Generate Proofscope URL node.	
	After choosing the type of note, the <b>Note window</b> opens and you can write your comment. You can use <b>Bold</b> , <i>Italic</i> or <u>Underline</u> text formatting.	
	You can add an extra area in the design to your note by selecting	
	Save Note saves the note and Remove Note removes it.	
	<b>Note:</b> After saving a note, it is no longer possible to remove it.	
	<ul> <li>Note: Selecting the Cancel icon on the top</li> <li>right of the Note window will not remove the note, it will only close the Note window.</li> </ul>	
	All notes are visible in the notes list on the <b>Sidebar</b> on the left if the <b>Notes view</b> is selected (by user or by page), and on the design itself. You can open the notes by clicking on the note in the Sidebar or by double clicking the visualization highlights in the design itself. Select <b>Save Note</b> to save your additional comment.	
	Notes can be edited if these 3 conditions are met:	
	<ol> <li>Notes are not in view only.</li> <li>The comment is the last one inside the note.</li> <li>The user is either the owner of the comment or an</li> </ol>	

Icon	Description
	The <b>Densitometer</b> allows you to measure separations. Select the exact spot in the file where you want to measure the inks and the information window appears: Black 4% Magenta 9% Yellow 10% Cyan 0% PANTONE 137 C 15% PANTONE 871 C 0% keyline 0% PACKZ INFOPANEL 0% PACKZ LOGO 0%
	Measure allows you to measure:
	<ul> <li>Measure allows you to measure:</li> <li>Vertical, horizontal and diagonal distances:</li> <li>Image: Constraint of the second second</li></ul>
	previews controlled by workflow <ul> <li>SETTINGS &gt; SETTINGS &gt; <ul> <li>QUANTUM &gt; Enable Distortion in</li> <li>Proofscope</li> </ul> </li> </ul>

Icon	Description		
	<b>Detect Barcodes</b> allows you to recognize and read barcodes. By dragging a rectangle on the barcode, the type and the value of the barcode is read:		
	CREEN TEA ORANGE		
	1D product	1D industrial	2D
	UPC-A	Code 39	QR Code
	UPC-E	Code 128	Data Matrix
	EAN-8	Code 93	Aztec
	EAN-13	Codabar	PDF 417
		ITF	MaxiCode
		RSS-14	
		RSS-Expanded	
	<ul> <li>Tip: You can select the value of the barcode and copy-paste it to another software tool.</li> <li>Attention: Barcodes should be 100% black or another dark separation. Do not use a mixed color space, as the screening will interfere with the barcode reading.</li> </ul>		e of the barcode software tool.
			be 100% black o not use a mixed will interfere with

**Note:** Note regarding *tile* sizes and zoom levels in PROOFSCOPE:

PROOFSCOPE stores four zoom levels (1x, 2x, 4x, 8x), with a maximum tile size of 2048 x 2048 pixels. The reason for this is to be able to store them in the database and to have reasonably sized tiles streamed to the browser. If the file is smaller than 2048 pixels at 96 dpi, the 1x tile corresponds to a 100% zoom, and you can zoom in 3 times. If the file is bigger than 2048 pixels at 96 dpi, the 1x tile corresponds to a <100% zoom (so the 1x fits in a tile), and you can zoom in 3 times.

# Generate report and download file

=

With these buttons you can generate an HTML report of the file or download the file.

Icon	Description
Q	<b>Generate Report</b> allows you to open the PROOFSCOPE report in HTML.

Icon	Description
D	<b>Download file</b> allows you to download the file in low resolution. In <b>Settings</b> you can specify if your file needs to be downsampled or rasterized.

# **View 3D files**

With PROOFSCOPE you can view 3D files.

You can view these types of 3D files in PROOFSCOPE:

- .ic3d files (IC3D).
- .dae files, which are XML files without graphics or other binary data (COLLADA).
- .zae files, which are compressed files containing the .dae file, a manifest file, graphics... (COLLADA).

To view a 3D file in PROOFSCOPE, open the file from **ASSETS**. In the PROOFSCOPE window you can move the 3D design around to view the different angles of the design. You can do this by holding down the left mouse button and moving around the mouse.

=

Note: All other options in the Sidebar and Toolbox are not available when viewing 3D designs.

# View screened files

With PROOFSCOPE you can view screened halftone files after a RIP process.

When viewing halftones, an extra tool becomes available in the toolbox:

**Measure Halftones** provides detailed information about the screens. It gives you a complete overview of the separations, angles, resolutions and percentages. If you drag a rectangle over the area of the file you want to measure, the information appears:



If you want to view/measure an individual separation, deselect all separations except the one you want to view in the **Information view** in the **Sidebar**:



The following halftone output files can be viewed with PROOFSCOPE:

- TIFF
- LEN

# **Related concepts**

Sidebar on page 304 On top of the left Sidebar, you can switch between the different view modes.

Toolbox on page 306

The Toolbox on top allows you to switch between different types of view tools.

# **Compare files**

With PROOFSCOPE you can compare files and view the differences.

To compare a file with another, follow these steps:

- 1. Select the first file from your assets and select the tab Compare File.
- 2. Select the file you want to compare the first file with and select Compare File again.
- **3.** PROOFSCOPE opens. Next to the options that were already available when viewing files, some extra options are now available:

Icon	Description
A	Represents the first file
Δ	<b>Show difference</b> highlights the differences between the two files
В	Represents the second file
A/B	<b>Switching Files</b> automatically switches between file A and B
<b>Z</b> B	Show Side by Side displays the files side by side

The comparison can be tweaked with the Differences view in the Sidebar (IB)

# **COMPARE VERSIONS**

This option is only available in case you work with different versions of the same file. To support versions in PROOFSCOPE, you need to open the **Generate Proofscope URL** node and set the **URL type** to **Folder as versions** or **Versions**.

- Compare: if you select this checkbox, you can compare different versions of the same file.
- Version A: here you can select the file that represents the first file (version A).
- Version B: here you can select the file that represents the second file (version B).

### FILE B

- Offset X: here you can select the vertical offset (in mm) of file B with respect to file A.
- Offset Y: here you can select the horizontal offset (in mm) of file B with respect to file A.
- **Rotation**: here you can select the amount of rotation of file B with respect to file A. Options are 0°, 90°, 180° and 270°.
- **Reset**: with this button you can reset the options to the default settings.

# **BOX FIT**

• **Box**: here you can select which box will serve as a reference point in case the files have a different position and you want to align them.

# AUTOMATIC FIT

You can use this option if you want to define a specific reference point on both A and B documents to align them.

- File A: here you can select the reference point on file A. Make sure file A is visible, select **choose**, position the cursor where you want to place the reference point on file A and click.
- File B: here you can select the reference point on file B. Make sure file B is visible, select **choose**, position the cursor where you want to place the reference point on file B and click. Next, select **Set** or **Calculate**.
- Set: if you select this option, the designs will be aligned exactly on the two reference points you have defined in File A and File B.
- **Calculate**: this button will use the reference point you have defined in File A and File B, but will use information of the images (such as colors) to align the designs more exactly.
- **Reset**: with this button you can reset the options to the default settings.

# ADVANCED

- Mode: if needed, you can select a different view mode to visualize the differences more clearly. Options:
  - **Default**: the default color mode will display the common pixels in File A and File B in white and the pixels that are not common in red. The pixels that are present in File A and not in File B are displayed in a darker red, the pixels that are present in File B and not in File A are displayed in a lighter red.
  - **Inverse**: in this case the common pixels will be displayed in black and the pixels that are not common will be displayed in red. The pixels that are present in File A and not in File B are displayed in dark red, the pixels that are present in File B and not in File A are displayed in lighter red.
  - Hard: in this case the common pixels will be displayed in gray. The pixels that are present in File A and not in File B will be displayed in blue, and the pixels that are present in File B and not in File A will be displayed in red.
  - **Darken**: in this case, both the common pixels and the pixels that are not common will be displayed in the original colors.
- Threshold: you can increase/decrease the threshold to see more/less differences

# A note on comparing and mapping separations

To define if separations are different in file A and file B, they are compared. In some cases, a separation in file B is mapped to a separation in file A.

The sequence of the mapping criteria is the following:

- 1. An exact match on the separations name.
- 2. A case free and unicode-composed free match on the separations name. Unicode has multiple representations for the same accented characters, and in this case, all notations are converted to the same notation before comparing.
- 3. In case only one technical colorant remains in A and B, they will be matched.
- 4. In case only one varnish colorant remains in A and B, they will be matched.
- 5. Finally, the LAB of the remaining colorants is calculated. The best match is paired first, then the second best match, etc. Matching is done if the delta E is lower than 10.

# Approve, reject or delegate files

With PROOFSCOPE you can approve, reject or delegate files in an approval workflow.

If a decision is required, one of the following options is displayed on the top right:

• To buttons: Approve or Reject.. This is the case when the approval was started without an Approval Form.



• A button **Show Approval Form**. This is the case when the approval was started with an Approval form. You can create an Approval Form in **Start Approval** node, parameter **Display Form**. If you select that button, the Approval Form is displayed in the Sidebar, where you can approve or reject the file:



If you have selected the checkbox **Allow Delegating** in the **Start Approval** node, a button **Delegate...** is also displayed. If you select this button, the decision to approve or reject the file will be delegated to another user. Options:

- Add approver: if you select this option, an extra participant of the approval workflow is added. This participant will receive a link to the file and will also be invited to approve, reject or delegate the file. Depending on the **Policy** in the **Start Approval** node, the file receives an *overall* status and is approved or rejected. Policy options:
  - All Participants need to Accept: all participants need to approve the file.
    - If all participants approve, the file is approved.
    - As soon as one of the participants rejects, the file is rejected.
  - All Participants need to assess: all participants need to either approve or reject.
    - If all participants approve, the file is approved.
    - If one of the participants rejects, the file is rejected, but only after all participants have assessed the file.
  - Only One Participant needs to Accept or Reject: the first participant determines the routing.
    - As soon as one of the participants approves, the file is approved.
    - As soon as one of the participants rejects, the file is rejected.
- Add informative approver: if you select this option, an extra participant of the approval workflow is added. This participant will receive a link to the file and will also be invited to approve, reject or delegate the file. However, this participant's decision is strictly informative and does not have an influence on the approval workflow.
- **Transfer approval**: if you select this option, the file is sent to another participant (who is not the initial approver). This participant will receive a link to the file and will also be invited to approve, reject or delegate the file.
- Approver e-mail: here you can enter the participant's email address.

See Approvals on page 298 for more information.

# **PROOFSCOPE** shortcuts

These are the shortcuts you can use in PROOFSCOPE.

	ú.	<i>î</i> :
Fit zoom	Cmd 0	Ctrl 0
Zoom in	Cmd +	Ctrl +
Zoom out	Cmd -	Ctrl -
Zoom in when in zoom mode	Click	Click
Zoom out when in zoom mode	Option + click	Alt + click

# Supported PROOFSCOPE formats

In PROOFSCOPE, the following formats are supported:

- JPEG
  - JPEG grayscale and RGB, with or without alpha channel.
  - CMYK
- TIFF
  - 1 bit, positive and negative.
  - Single channel, positive and negative, with or without a single alpha channel.
  - RGB, with or without a single alpha channel.
  - CMYK, with or without a single alpha channel.
  - Multi-channel, with or without a single alpha channel
  - For multi-byte there is currently only support for interleaved data, not planar (TBA), regular and striped.
  - For compressions there is support for support none, pack-pits, LZW and JPEG.
  - Support for decompressing the CCITT3 and CCITT4.
  - No support for indexed color.
  - For alpha there is support for both pre-multiplied and normal.
  - Support for combined viewing of a set of TIFF files.
  - Color info from the ITF\_CMYK tag.
  - Info from embedded XMP data.
  - Info from the Adobe Photoshop blob (for example color info and channel types).
- PDF
  - Most types of PDF files.
  - No interactive forms.
  - No media (movies etc.).
  - No 3D PDF.

**Note:** PROOFSCOPE renders any PDF/X-4 file correctly without the need of extra configuration. None of the PROOFSCOPE options regarding file rendering influences how PDF/X-4 files are rendered.

- LEN
  - Only 1 bit files, which is the output of the RIP process.
  - Just like TIFF files, LEN files can be viewed in combination.
- BMP
- CF2

• EPS

# **PROOFSCOPE SDK**

The PROOFSCOPE SDK provides you with the tools and documentation to integrate PROOFSCOPE in other software systems.

It contains the following parts:

- The CLOUDFLOW API, which provides you with the calls needed to integrate PROOFSCOPE into your system.
- The PROOFSCOPE component, a jQueryUI widget which allows you to easily build the PROOFSCOPE interface.

# The CLOUDFLOW API

The CLOUDFLOW API provides you with an overview of the calls needed to integrate PROOFSCOPE into your system.

### Authentication

You need a security context to make API calls to CLOUDFLOW. With this API call you can create a session key.

The call api.auth.create\_session(user\_name, user\_pass) creates a session key which can be used in the other API calls. A session key created with this call expires after 12h.

### Example

```
The call api.auth.create session('admin', 'admin')
```

returns:

```
{ session:
"553d66a7a32901000000091F7AFE29CECEA99322E1F31BC5D11BC321435874639"}
```

To use this session for other API calls, it can be set on the API object this way:

```
api.m_session =
"553d66a7a32901000000091F7AFE29CECEA99322E1F31BC5D11BC321435874639"
```

### Import a file in CLOUDFLOW

With this API call you can import a file in CLOUDFLOW.

api.file.copy file(source path, destination path)

Both parameters use the URL schema. As the imported file will be located outside CLOUDFLOW, a file:// schema will be used. The destination path will be a CLOUDFLOW URL.

### Example

In Windows, you can import c:\Files to import\document.pdf in CLOUDFLOW with this API call:

api.file.copy\_file("file:///c:/Files%20to%20import/document.pdf",
"cloudflow://PP FILE STORE/Demo%20Files").

You can access document.pdf in CLOUDFLOW with this URL: cloudflow://PP\_FILE\_STORE/Demo %20Files/document.pdf.

On , you can import /Users/joe/Files to import/document.pdf (located on the Macintosh HD drive) with this API call:

```
api.file.copy_file("file:///Macintosh%20HD/Users/joe/Files%20to%20import/
document.pdf" "cloudflow://PP FILE STORE/Demo%20Files").
```

It returns the CLOUDFLOW URL of the imported file:

```
{
   copied_file: "cloudflow://PP_FILE_STORE/Demo%20Files/document.pdf"
}
```

### Remove a file from CLOUDFLOW

With this API call you can remove a file from CLOUDFLOW.

api.file.delete\_file(path)

The CLOUDFLOW URL of the asset can be passed.

### Example

```
The call api.file.delete_file("cloudflow://PP_FILE_STORE/Demo%20Files/
packaging crab final.pdf")
```

removes the asset and file packaging crab final.pdf from the Demokit files.

### Get file metadata

With this API call you can retrieve an asset metadata.

api.asset.get by url(cloudflow url)

### Example

If you need the information of the asset with URL: cloudflow://PP\_FILE\_STORE/Demo%20Files/ Cakepops NEW.pdf

the API call:

```
api.asset.get_by_url("cloudflow://PP_FILE_STORE/Demo%20Files/
Cakepops_NEW.pdf")
```

returns:

```
{
" id": "553a171ce7c4000000004a1",
"url": "PP FILE STORE/Demo Files/Cakepops NEW.pdf",
"sub": "",
  "cloudflow": {
  "file": "cloudflow://PP FILE STORE/Demo%20Files/Cakepops NEW.pdf",
  "enclosing folder": "cloudflow://PP FILE STORE/Demo%20Files/",
  "part": "cloudflow://PP FILE STORE/Demo%20Files/Cakepops NEW.pdf"
},
"file name": "Cakepops_NEW.pdf",
"file extension": ".pdf",
"document name": "Cakepops NEW",
"path": [
  "PP FILE STORE",
  "Demo Files"
],
"filetype": "application/pdf",
"modtime": 1426057808,
"quantum": {},
"mime types": [
  "application/pdf"
"file time": {
  "creation local": "2015-04-24T09:53:19+0200",
  "creation utc": "2015-04-24T08:53:19Z",
  "creation": 1429858399,
"modification_local": "2015-03-11T08:10:08+0100",
  "modification utc": "2015-03-11T07:10:08Z",
```

```
"modification": 1426054208
},
"file_size": 43091942,
"pagecount": 1,
...
}
```

# Create a view URL

With this API call you can create a view link for the PROOFSCOPE viewer.

```
api.proofscope.create_view_file_url_with_options(host_url, file_url, options)
```

Parameters:

- **host\_url**: the public host URL where CLOUDFLOW is accessible (for example http://my.company.com/).
- file\_url: the CLOUDFLOW URL of the asset to view (for example cloudflow://PP\_FILE\_STORE/ viewable%20files/afile.pdf).
- options: a JSON dictionary with optional parameters.
  - email: the email of the user that will view the file.
  - time\_out: the amount of seconds the link should be available, by default 14 days.
  - **Note:** This call just returns an URL, it doesn't send an email to the user. If the contact doesn't exist, it will be created on the fly.

### Example

=

If you want to view the asset cloudflow://PP\_FILE\_STORE/Demo%20Files/Cakepops\_NEW.pdf, and if you want to make it available for the user with e-mail: john@...

the API call:

```
api.proofscope.create_view_file_url_with_options(
   "http://localhost:9090",
   "cloudflow://PP_FILE_STORE/Demo%20Files/Cakepops_NEW.pdf", {
   email: "john@domain.com"
})
```

returns:

```
'"result": "ok",
"url": "http://localhost:9090/portal.cgi?
proofscope&asset_id=553a171ce7c4000000004a1&email=john
%40domain.com&temp_scope=1471c75d-b153-4b0d-90a6-
cd8284447605&signature=82aea0510f246eff788a77dbdf7f70b3"
}
```

# Create a diff URL

With this API call you can create a difference view link for the PROOFSCOPE viewer.

```
api.proofscope.create_view_file_difference_url_with_options(host_url,
file_url, diff_url, options)
```

Parameters:

- host\_url: the public host URL where CLOUDFLOW is accessible (for example http://my.company.com/)
- file\_url: the CLOUDFLOW URL of the asset to view (for example cloudflow://PP\_FILE\_STORE/Demo %20Files/packaging\_crab\_orig.pdf).
- diff\_url: the CLOUDFLOW URL of the asset to diff with (for example cloudflow://PP\_FILE\_STORE/ Demo%20Files/packaging\_crab\_corrected.pdf).

- options: a JSON dictionary with optional parameters.
  - **email**: the email of the user that will view the file.
  - time\_out: the amount of seconds the link should be available, by default 14 days.
- **Note:** This call just returns an URL, it doesn't send an email to the user. If the contact doesn't exist, it will be created on the fly.

### Example

```
If you want to view the difference between the files cloudflow://PP_FILE_STORE/Demo%20Files/
packaging_crab_orig.pdf and cloudflow://PP_FILE_STORE/Demo%20Files/
packaging_crab_corrected.pdf
```

the API call:

```
api.proofscope.create_view_file_difference_url_with_options(
   "http://localhost:9090",
   "cloudflow://PP_FILE_STORE/Demo%20Files/packaging_crab_orig.pdf",
   "cloudflow://PP_FILE_STORE/Demo%20Files/packaging_crab_corrected.pdf", {
    email: "john@domain.com"
})
```

returns:

```
{
    result: "ok",
    url: "http://localhost:9090/portal.cgi?
    proofscope&asset_id=553a171ce7c4000000004e0&email=john
%40domain.com&temp_scope=9d15496c-4651-4916-b56d-
d340b39f6961&view_id=41d768b7-94cd-4aa6-9356-
c558cbf5555e&signature=affb1cba796f20c0243e174c24707404"
}
```

### Retrieve a thumbnail of an asset

With this API call you can retrieve a thumbnail of an asset, which can be used in an **img** tag in an HTML page.

```
http://<server>:<port>/?asset=get_thumbnail&file=<cloudflow url>&page=<page
number>
```

Parameters:

- **cloudflow url**: the CLOUDFLOW URL of the asset.
- page number: an optional parameter, 0 is the first page.

For the asset cloudflow://PP\_FILE\_STORE/Demo%20Files/Cakepops\_NEW.pdf, this is the thumbnail URL of the first page:

```
http://localhost:9090/?asset=get_thumbnail&file=cloudflow://PP_FILE_STORE/Demo
%20Files/Cakepops_NEW.pdf
```

# Customize user preferences

With these API calls you can customize several user preferences.

# Introduction

You can customize the following PROOFSCOPE settings for a user:

- Units
- Language
- Note color
- UI customization

For these settings PROOFSCOPE uses the CLOUDFLOW preferences.

### **Structure of Preferences**

The CLOUDFLOW preferences are divided in two realms:

- System preferences (system)
- User preferences (**user**)

User preferences inherit from the system preferences. For example, if a preference key **language** is defined on system level, it is inherited on user preference level if it is not present there. The idea behind it is that a CLOUDFLOW system defines the defaults and that the users can override these defaults. User preferences are bound to the email address of the users.

You can assign further preferences per application. To bind a preference to an application an identifier is used, called the application key (for example **com.nixps.proofscope**). You can leave the application key empty, in which case it not specified for an application.

This document describes how to use the API to change the preferences for a specified user. If you want to retrieve or save your own preferences, you need to use other calls.

# **Preferences API**

You can change a CLOUDFLOW preference with the save\_for\_realm API call:

You can get all of the preferences of a user with get\_for\_realm API call:

```
api.preferences.get_for_realm(
   'user', // the realm to use, here 'user'
   user_email, // the email for the user
   '', // the application key, here empty
   '' // the key of the preference, empty returns everything
);
```

The user\_email parameter should be the same as the one used for creating the view URL.

Here's an overview of preference keys used in PROOFSCOPE:

(Preferences) Key	Application Key	Description
units.length	••	The unit for lengths
units.small_length	••	The unit for small lengths
units.ruling	••	The unit for rulings
language	••	The language in the UI
noteColor	com.nixps.proofscope	The color of the PROOFSCOPE notes
hideSidebar	com.nixps.proofscope	Hide/show the sidebar when opening PROOFSCOPE

(Preferences) Key	Application Key	Description
defaultPanel	com.nixps.proofscope	The default side panel to show when opening PROOFSCOPE

# Setting the units

PROOFSCOPE uses three unit types:

- length: used for showing distances, for example measure tool.
- **small\_length**: used for showing small distances.
- **ruling**: used by the measure halftones tool.

A unit has two fields:

- accuracy: digits after the decimal.
- **unit**: the unit to use (see units table).

Units table:

Unit	Unit name	Usage
pt	points	length, small_length
in	inches	length, small_length
mm	millimeters	length, small_length
cm	centimeters	length, small_length
m	meters	length, small_length
dpmm	dots per millimeter	ruling
dpi	dots per inch	ruling
lpmm	lines per millimeter	ruling
lpi	lines per inch	ruling

You can set the length of the unit with this API call:

```
api.preferences.save_for_realm(
  {accuracy: 3, unit: 'in'},
  'user',
  'john@domain.com',
  '',
  'units.length')
```

This will set the length unit to inches with 3 digits after the decimal.

In the following example the small\_length unit is set to inches, with five digits after the decimal:

```
api.preferences.save_for_realm(
  {accuracy: 5, unit: 'in'},
  'user', 'john@domain.com',
  '',
  'units.small length')
```

In the following example the unit for the ruling is set to DPI:

```
api.preferences.save_for_realm(
  {accuracy: 0, unit: 'dpi'},
  'user',
```

```
'john@domain.com',
'',
'units.ruling')
```

### Setting the language

The language preference contains the ISO code of the language to use in the PROOFSCOPE UI.

In the following example the language is set to English:

```
api.preferences.save_for_realm(
    'en',
    'user',
    'john@domain.com',
    '',
    'language')
```



Note: The ISO 639 two digits language codes are used to select a language (in this example, en for English)

### Setting the note color

A user can have a specific note color. There are several predefined note colors:

- yellow
- green
- blue
- purple
- orange
- pink

You can also specify an HTML note color, for example **#FF0000** for red.

In the following example the note color is set to green:

```
api.preferences.save_for_realm(
    'green',
    'user',
    'john@domain.com',
    'com.nixps.proofscope', // Make sure to specify this application-key
    'noteColor')
```

### Show/hide the side bar

You can control the visibility of the side bar when opening PROOFSCOPE. To do this, you can set a boolean flag called hideSidebar to **true** or **false**.

Here is an example that will hide the side bar when opening PROOFSCOPE:

```
api.preferences.save_for_realm(
    false,
    'user',
    'john@domain.com',
    'com.nixps.proofscope', // Make sure to specify this application-key
    'hideSidebar')
```

#### Set the default side panel

You can set the default side panel that the user will see when opening PROOFSCOPE. You can set the preference key defaultPanel to thumbnails, notes or separations.

Here is an example that will set the default side panel to the separations panel:

```
api.preferences.save_for_realm(
    'separations',
    'user',
    'john@domain.com',
    'com.nixps.proofscope', // Make sure to specify this application-key
    'defaultPanel')
```

# The PROOFSCOPE component

With the PROOFSCOPE component you can easily build the PROOFSCOPE interface.

As described in Create a view URL on page 318 and Create a diff URL on page 318, PROOFSCOPE can be started as a page served from CLOUDFLOW.

However, PROOFSCOPE can be further customized. This can be useful in two cases:

- In case you want to integrate PROOFSCOPE in a custom HTML page, which can be hosted on another web server.
- In case you want to integrate PROOFSCOPE in external an approval system.

You can do this using the PROOFSCOPE component. This is a jQueryUI widget, which has a standard API to its components. See http://jqueryui.com/ for more information on jQueryUI and how the API works.

When bootstrapping PROOFSCOPE you will need a viewer token, which consists of a view URL constructed with **create\_view\_file\_url\_with\_options**. When using PROOFSCOPE with such a token, it will use all parameters passed with the **create\_view\_file\_url\_with\_options**.

# Case 1: integrate PROOFSCOPE in a page

This case describes how you can integrate PROOFSCOPE in a page.

The PROOFSCOPE SDK contains two HTML files which contain the base to start a PROOFSCOPE component on a custom HTML page:

• **createToken.html** contains the code to create a view token. This token should be saved and passed when the viewer needs to be opened.

You can find it here: http://<cloudflow server>/manual/ProofscopeSDKExample/ ProofscopeComponent/createToken.html

• **proofscope.html** contains the code to integrate the PROOFSCOPE component. It shows a prompt dialog where you can input the viewer token.

You can find it here: http://<cloudflow server>/manual/ProofscopeSDKExample/ ProofscopeComponent/proofscope.html

You can copy this code of proofscope.html:

```
1 api_async.proofscope.get_view_info(pToken, function(pResult) {
2     api.m_session = pResult.use_session;
3     api_sync.m_session = pResult.use_session;
4     api_async.m_session = pResult.use_session;
5     
6     var viewerParameters = pResult.view.parameters;
7     $('#proofscope').Proofscope(viewerParameters);
8 });
```

• Line 1 will make the call to get the PROOFSCOPE parameters that were passed when creating the token.

- Line 2 to 4 will set the session which is returned, so the internal functions of PROOFSCOPE will work with the user for which the token is generated.
- Line 6 will retrieve the PROOFSCOPE parameters.

• Line 7 will construct the PROOFSCOPE viewer in a div with id **proofscope**. It will take the parameters that were used to create the token.

### Case 2: integrate PROOFSCOPE in an external approval system

This case is an extension of case 1 and describes how to integrate PROOFSCOPE in an external approval system that keeps track of approvals.

In this case you can show the approval state of the assets in the external system.

```
1 api async.proofscope.get view info(pToken, function(pResult) {
      api.m session = pResult.use session;
2
3
      api sync.m session = pResult.use session;
4
      api async.m session = pResult.use session;
5
6
      var viewerParameters = $.extend(pResult.view.parameters, {
7
          standalone: {
8
9
              approve: true,
10
              getAssessment: function(pSession, pFileInfo, pSuccess,
11
pFailed) {
12
                  pSuccess (assessment);
13
              },
14
15
              setAssessment: function(pSession, pFileInfo, pAssessment,
pSuccess, pFailed) {
16
                  assessment = pAssessment;
17
                  pSuccess();
18
              }
19
          }
20
      });
21
      $('#proofscope').Proofscope(viewerParameters);
22
23 });
```

- Line 6 uses the **\$.extend jQuery** function (see https://api.jquery.com/jquery.extend/). This function will merge two JSON structures together and returns the merged results. It will add the JSON of lines 7 to 19 to the viewer parameters that were returned by the get view info call.
- Line 7 will start a JSON structure called **standalone**. This JSON contains parameters specific to PROOFSCOPE integrations outside CLOUDFLOW.
- Line 9 will set the flag **approve** to **true**. It instructs not to use CLOUDFLOW approvals anymore, but to use an external system for approvals.
- Line 11 will set the **getAssessment** callback, which is called by PROOFSCOPE. It should return the current approval state. Here you will need to enter the integration code with the external system. You receive the session information (first parameter) and the file information (second parameter). When you have received the information the **pSuccess** function is called (line 12). It takes one parameter: the current assessment. The current assessment is a string which can be:
  - **pending**: in this case the user needs to approve this file.
  - **accept**: in this case the user has approved this file.
  - reject: in this case the user has rejected this file.
  - **none**: in this case the user doesn't need to approve this file.

If the external system returns an error (for example **connection error**, **unavailable**...), you can call **pFailed** with an error description. For example **pFailed**("**Could not connect to ...**").

• Line 15 is similar to line 11, but will send the assessment from PROOFSCOPE to the external system when the user clicks the **approve** or **reject** button. The assessment is encoded in **pAssessment** and is either **accept** or **reject**. Session and file information is passed in the first and second parameter. You can call the **pSuccess** or **pFailed** functions when the external back end is updated.
• Line 22 will create the PROOFSCOPE component with the described setup.

# The Session object

The session object has two functions. Here is an example:

```
1 getAssessment: function(pSession, pFileInfo, pSuccess, pFailed) {
2   var email = pSession.getUserEmail();
3   ...
4   pSuccess(assessment);
5 },
```

Line 2 returns the users email for the session and stores it in the variable email.

There are two functions for the Session object:

- getUserID: this function returns the ID of the user for the PROOFSCOPE session.
- getUserEmail: this function returns the email of the user for the PROOFSCOPE session.

#### The Fileinfo object

The functionality of the Fileinfo is identical to Session object.

There is one function for the Fileinfo object:

• getAssetID: this function returns the CLOUDFLOW asset ID that is currently viewed.

# PACKZflow

# PACKZflow

With PACKZflow you can automate prepress operations in CLOUDFLOW.

PACKZflow works together with CLOUDFLOW WORKSPACE:

- It has its own category of nodes.
- It can work with variables and files.

See PACKZflow nodes for more information on the PACKZflow nodes.

# PATCHPLANNER

PATCHPLANNER is a web-based production tool for the corrugated plate industry to generate patches and optimize your platemaking process.

**Note:** Availability:

The **PatchPlanner** UI is not available for Tectonics customers except when expressly requested and configured.

When producing corrugated boxes, in many cases only a small area needs to be printed although the box itself can be very large. To avoid too much waste of expensive flexo plates, PATCHPLANNER separates the layout of the printed areas into individual elements (**patches**) so that a much smaller plate can be used than the actual size of the job.

PATCHPLANNER automatically creates patches and positions them on a sheet in a way they take up as little space as possible. This data is exposed on a small plate and if needed, sent to a cutting machine. Next, the patches are mounted on a carrier sheet.

PATCHPLANNER works with **Jobs** (creating the patches) and creates **Sheets**, a combination of Job Patches on a sheet in order to give you optimum plate and film material usage.

# Create a job

In a PATCHPLANNER context, a job is one file that is divided into patches.

To create a job, open PATCHPLANNER in CLOUDFLOW and follow these steps:

#### 1. Create a new job

- 1. Select New....
- 2. Select Job.
- 3. Enter a job name and select Create.

# 2. Upload a PDF file

- 1. Select Upload.
- 2. Select Browse files... and select the file(s) you want to upload.
- 3. Select the Mounting method. Options:
  - Automatic (MOM): select this option in case the patches will be mounted automatically on the carrier in the mounting machine. A MOM XML file that contains the coordinates of the patches will be exported .
  - **Mirror**: select this option in case the mounting of the patches will be done by mirror mount. A mirror proof PDF will be exported. This file is printed and taped onto the proofing drum of the mounting machine. It will display through a mirror and the marks allow for the patched job to be positioned exactly.
  - Drill Mount: select this option in case the mounting of the patches will be done by drill mount.
  - Mirror and Automatic (MOM): select this option if you want to combine the two options above.
- 4. Select Close

#### 3. Change the job settings

The jobs are previewed in the left hand panel and contain two red badges. The left badge indicated that there are no cylinder marks applied and the right badge indicates that distortions have not been set.

To change the job settings, select the Info button next to the preview.

- **Review File**: this will open the file in **PROOFSCOPE** and allows you to perform a quality check before creating the patches.
- Slug Line
  - Base name: depending on the output method, the Slug Line Base name has a different functionality:
    - When you selected **Mirror** during file upload, the Slug Line Base name will be the name of the MOM XML file.
    - When you selected **Automatic (MOM)** during file upload, the Slug Line Base name will be the name of the mirror proof file output. The output is a PDF file containing the specified die line, which separation is set under the **Mirror Mounting** Job setting.



#### • General

- **Distortion**: here you can set the distortion that will be applied on the output.
  - **Note:** Artwork is sent to PATCHPLANNER undistorted. Therefore, distortions must be applied in PATCHPLANNER to send appropriate coordinates to the mounting machine.
- Mirror Mounting (only available when you selected Mirror during file upload):
  - Separation: here you can select the separation of the die line which will be added to the mirror proof file output.
  - **Distortion**: here you can set a distortion in order to adjust the patch marks to the proof cylinder compensating for the difference between the mirror mount cylinder and the printing cylinder.
- Drill Mount settings (only available when you selected Drill Mount during file upload):
  - Die: here you can specify the die separation.
  - Carrier box: here you can specify the predefined PDF box.
- Cylinder (only available when you selected Automatic (MOM) during file upload): here you can enter information about the cylinder circumference and width for faster job creation.
- Thickness (only available when you selected Automatic (MOM) during file upload): here you can enter the thickness of the plate, mylar, paper and tape. This information is sent to the mounting machine to adjust the pressure for proofing, mounting and between the rollers.
- Marks: here you can select patch marks. You can select default or custom marks that were uploaded to PATCHPLANNER.

To upload marks, follow these steps:

- 1. Close the job settings window in case it's open.
- 2. Select Settings... in the job window.
- 3. Select Browse files... in the Upload window.
- **4.** Select the marks file.
- 5. Select Save.

The newly uploaded marks are now added to the list in the job settings.

- Patch Margins: here you can select margins to set to the patches.
- Cylinder Marks (only available when you selected Automatic (MOM) during file upload): here you can set cylinder marks which allow you to set coordinates that are not in the artwork to relay to the mounter.

**Use case**: in many cases corrugated plate manufacturers use press registration marks that were imaged on *scrap* material. Indicating cylinder marks is a way to relay those coordinates to the mounter to apply these *loose* marks in perfect register. The advantage is that this takes less plate material and fewer mark shavings after mounting.

To set cylinder marks, follow these steps:

- 1. Select Set Cylinder Marks...
- 2. Select the + icon to add the marks.
- 3. Click on the location where you want to position the marks.
- 4. If needed, drag the marks to the correct position.

To delete the cylinder marks, select them and select  $\widehat{\mathbf{m}}$ .

When you close the job settings the job is ready to be patched.

#### 4. Patch the file

1. Double click on the job preview icon to launch the patching mode.

2. Select **Detect Patches**. PATCHPLANNER uses an algorithm that automatically detects patches. It analyzes all the colors of the document and creates patch areas for least plate usage.

There are two different patch colors:

- a. Blue patches mean everything is OK to mount.
- **b.** Red patches can mean one of the following things:
  - The patch is too small to mount with both cameras.
  - One or more patches intersect with another patch.
  - The patch clips artwork (although this should never occur after applying Detect Patches).

You can adjust these patches with the patch tools:

- Smart Patch Selection (): when you draw a selection rectangle around a part of a patch, this tool will detect and select a complete patch.
- Select Patches (): with this tool you can select a patch by drawing a selection rectangle around it. When a patch is selected, you can:
  - Make it bigger or smaller by positioning the cursor on one of the four sides and move the arrow in the desired direction.
  - Delete it by selecting the X in he upper right corner.
- Merge Patches (): with this tool you can merge several patches by drawing a selection rectangle around them.

Note: The Smart patch tools remain inactive (grayed) until all of the patch data has been loaded.

3. After patching, select Close. This will position all patches on a layout sheet.

#### 5. If needed, edit the patch marks

In this stage you can add, reposition or delete patch marks. To do this, follow these steps:

- 1. Select the patch you want to change the marks for.
- 2.

=

- Select the Edit patch marks icon (
- To reposition the marks, select one of the marks and drag it to the desired position.
- To delete the marks, select one and select the delete icon (
- To add marks, select one of the add marks icons:
  - Add marks (1): select this option if you want to add marks that will just help you positioning the plate.
  - Add main marks (): select this option if you want to add main marks. Main marks are used during automatic positioning on the carrier. Their coordinates are added to the MOM XML file. Main marks contain a number which tells the operator when to put the patch on the mounting machine.
- **Note:** You can preview the patch both in **Separated** and **Composite** mode.
- **Note:** In case of drill mounting, each mark can be positioned independently of each other without the restriction to position them at the same height or width.

#### 6. Commit Job Patching

To complete the job patching process, select **COMMIT JOB PATCHING**. Your job is now ready to assign to a sheet.

# Create a sheet

A sheet in PATCHPLANNER represents the flexo plate. It will be filled with patches in the most optimal way.

To create a sheet and fill it with patches, follow these steps:

#### 1. Create a sheet with the correct dimensions

- 1. Select New....
- 2. Select Sheet.
- 3. Enter a sheet name and select Create.
- 4. Select Sheet size.
- 5. Enter the Patch Sheet Size and Sheet Margins.
- 6. Select Save.

# 2. Add a job

- 1. Select Add job.
- 2. Select the job you want to add and select Add. The total number of patches placed on the job are displayed on to the right of the job name. For example, 0 of 14 means that 0 of 14 total patches have been placed onto a sheet for output. When you add patches to the sheet, this number will increment.
- 3. Repeat if you want to add another job.

# 3. Fill the sheet with patches

When you have added the jobs, a preview of the job patches is displayed on the left. You can add patches to the sheet in two ways:

- Manually by selecting the patch icon on the left and drag it to the sheet.
- Automatically by selecting Calculate layout. This functionality places the unplaced patches on the sheet in the most optimal way. The patches that were already placed do not move.

The placed patch icons will show in green and the unplaced patch icons will show in white.

The following tools are available:

Icon	Functionality
۵ ۵	Deletes the selected patch
5 C	Rotates the selected patch
5 C	Snaps the selected patch to other patches when moving to prevent overlapping
	Clears the sheet

#### 4. Export

When the sheet is filled with patches in the most optimal way, select **Export to PDF**. This will export the following PDF files:

- A black color PDF file that can be RIPped on any RIP for output. The screening and angles are maintained from the original screening set in the PDF file.
- A PDFOverview file, which serves as a *cheat sheet* for the cutting department to enable them to immediately see which patches belong to which job.

Only in case you selected an automatic cutter in SETTINGS > PATCHPLANNER > FORMAT OUTPUT FOR
 > cutting method: a PDF file with the cutting information for a Zund or Kongsberg cutting table.



**Note:** To get the bevel knife to work appropriately, the PATCHPLANNER output PDF file for cutting needs to be quickly opened in ArtiosCAD and saved as an ACM file.

# RIP

CLOUDFLOW RIP converts graphical artwork to 1 or 8 bit TIFF raster files which support all printing processes.

#### Overview of the specifications of the CLOUDFLOW RIP

- Support for Flexo platemaking. Therefore, sophisticated screening and calibration control is provided.
- Support for the following types of screens:
  - AM or HPS screens.
  - FM or HDS screens, a second-generation stochastic screen.
  - HXM screens: a combination of AM and FM screens.
- Object based screening, which enables you to have full control over the RIP process.
- Support for full calibration control on both press and print curves:
  - The press curves allow the correct application of bump-ups/clamping, in both highlights and shadows.
  - These curves can be assigned per separation and per screen.
- TIFF output generation. An integrated dot viewer allows you to inspect of the generated output, with tools to measure density, angles and LPI, in both a separated or composite way.
- An integrated dot viewer which allows you to inspect the generated output, with tools to measure density, angles and LPI, in both a separated or composite way. See View a file with PROOFSCOPE for detailed information on the dot viewer.

#### Modes

The CLOUDFLOW RIP comes in two modes:

- Standalone. In this case the RIP comes with a separate web-based UI in which you can set up presets and run RIP jobs. See Standalone RIP on page 349 for more information.
- As a workflow node. In this case the RIP runs as part of a real workflow. See the RIP node for more information.

# Screening

CLOUDFLOW RIP supports the following types of screens:

- AM (HPS) screens.
- FM (HDS) screens, a second-generation stochastic screen.
- Hybrid (HXM) screens: a combination of AM and FM screens.

#### AM screens

AM screens or HPS (Harlequin Precision Screening) provide more accurate screening and reduces moiré without unduly limiting the choice of screen angle and frequency.

HPS allows you to select any screen frequency and to use the usual CMYK screen angles of  $0^\circ$ ,  $15^\circ$ ,  $45^\circ$ , and  $75^\circ$  (plus multiples of  $90^\circ$ ).

To reduce moire patterning, HPS uses an adaptive screening technique that can adjust each halftone dot so that it is placed within one half pixel of its ideal location.

In some cases, you can override a set of angles requested in a job. This is especially useful if the job has requested a set of angles that optimize the output quality for a particular output device, such as a laser printer, but that may diminish the quality on other devices, such as a plate setter. You may want consistency. For example, on all pages of a single publication when the jobs come from different sources. Enforcing settings in the RIP is the simplest way of getting such consistency.

A good choice of angles for general use with color separations in offset litho work is a set in which the colors are separated by 30°, for example: 15°, 75°, 0°, and 45° respectively for CMYK (and related sets using these angles plus or minus multiples of 90°). For use with **elliptical** dots, a separation of 60° is recommended, leading to angles of 15°, 75°, 0°, and 135° for CMYK.

Using a **Euclidean** dot shape produces better saturated grays at gray values above 50%, especially at finer (higher) screen frequencies. Euclidean strategies increase the fill of halftone cells from the corners, instead of the centers, when the gray value exceeds 50%. That is, when the gray value is less than 50%, the dots are black, the background is white, and the dot size increases as the gray value increases; when the gray value reaches 50%, the dots become white, the background becomes black, and dot size decreases as the gray value increases.

Manipulating the dot shape generated can greatly influence the amount of dot gain in an image. CLOUDFLOW RIP offers a variety of dot shapes, and your choice between them might depend on the output device, resolution as well as media being used. Standard AM Dot shapes included with the RIP are:

#### Cosine Dot

	•	٠	•	ᄀ	•	•	•	1	,	•	•	٠	٠	٠			٠	٠	٠				•		-												Ð							•	(	•	•	•
1		•	•		•	•	•	•	٠	•			•	•	•	•	•		•	•	•		•			•			-	- e	<b>—</b>							Т			th –					•		
	•	•	•		•	•	•	4 .	•	•	•	•	•	•			•	٠	٠	le			•	- el			) <b>(</b>							П	T	1	ID T									•	•	•
1		•	•		•	•	•						•	•					•	•					•	•			- <b>•</b>	- e					П						th –							
		•			•	•	•	4 .		•	•	•		•			•	٠			•		•		-	- <b>•</b>	- <b>1</b>	- T				1			Y		i D					1			4	•	•	
1.		•	•		•	•							•	•	•					•	•	- e	- e		• -	•			-	٠.				71	n			T	1		th T		Ξ.		•			
	•	•			•	•	•			•	•	•	•	•			•	٠	•		- T	•	•		-	- ۲	) <b>–</b> I	1	-	•					¥			П		Тn		11	• 1	•	d -	•	•	•

Cross



#### Diamond

• Double



• Double Dot

				-			_	_					-	[		-	-	-		-	-							Γ				ᅮ									-																			<b>.</b>													-					2
			•				· 1	•			•		- 41	•	٠	•			-41	•	•	•	•		•	1.	٠	٠	•	•	•	- 110																										16																				
							- I	ŀ					• II		•				. 16	•	•	٠	٠	•	•	•					•				•					_	_				πГ							In the	n							CT T																		
							· I	•					- 41						- 41			•	•							•	•				-			•1		-	-	_	_	_				10	10	10	10		n		<b>—</b>	_					-			- 11														
							- I	ŀ					• II		•				. 11	٠.	٠.	٠.	٠.	•	• 1	1.	•			٠.	٠.	· II -		•	•	• •		- 1	<b>-</b>	-	- 11		10		ΠГ	b D	<b>1</b> 0	o	o		~	in.	<b>~</b>	~		-	~	10	-	~					1.1					-								
							· I						- 41						- 41	•	•	• •	• •						•	•	•				-	- e i		- 1		-						_		-	-				-	~	_	_		11		-	-																	
							- I	ь÷.					• II		۰.	۰.	•	•	. 11	۰.	٠.	٠.	٠.	•	• 1	1.	• •			٠.	٠.	-11-		•	•	• •		-1	⁻.	-					чΠ	10	10	0	-		<b>C</b> 1	10.	~	~	Υ.	-	~	10	-	~			_	-						-								
			1 ·				· I	۰.					- 41	•	•	•			-41	•	•	• •	• •	• •	÷.	•			•	•	•	- II e		-	- e	-		•	•	• •					11.	C C C	<b>.</b>	ъr	Transition and the second s	200	201	11	'n	$\sim$	~	-	~	117		-	-																	
							- I	ŀ					• II	L .					. 11	•		•	•	•	•	1.1	• •			٠.		· II -		•	•				<b>-</b>	_		-			10	•					<b>m</b>	in.	~				-			~		_		-														
							. 1						- 41					÷ .	. ЛI	•	•	• ` •	•	· .	· •			•	•	•	•			-		-		-1	<b>H</b> -1		_					<u></u>	_	-	-	20	201	11-	~	~	<u> </u>	-	~	110		-	-	-	-															
	1.11			· • `			. I	ь÷.	• •	· .	· • ·		. 11	L .	۰.	۰.	· • ·	•	. 11	۰.	٠.	٠.	٠.	•	• 1	L۳.		· .	۰.	۰.	۰.	-11-			•	• •		-1	<b>-</b>		- 11				10	20	20	0	0	-	<b>.</b>	10-	~	~	Υ.	-	~	10	-	~	-	-			11			-	-	-								
							. I						÷ л			- °		÷.,	: JI	•	• *	۰° .	• •	· .	· •	I'		•	•	• "	• -							-1								~	-		- 10	202	201	11-	~	~	~	_			~	-	-		-	-														
				i			· . I	L .	· ` .	· .	· • ·		. 11	L .	۰.	۰.	· . ·		. 11	۰.	٠.	•	۰.		• 1	L.,		٠.	· ·	۰.	۰.	· II -			•	• •		-1	г.							×	60	0	0	0	<b>~</b> ~	10-	~	~	~	-	~	- 11	-	~	-	_			1.	-	-			-								
11.1.1.1.		1 . T	1 . T		÷.,		. ' I	Γ.	1.1				: JI	r.'	. '	. "	· ' .	÷	· JI			٠Ť.		. × .	· .	I'	• • •					-11-		-		'		-1								<u></u>	-	~	×*	× .		11.	~	~	<b>-</b>		~	10		-	-	-	-	-		-		× .	-			т.		-	-	-	-	-
		_	_	_	_	-	_	_	-	-	-	_	_	-	-	-	_	_	_	-	-	-	-	_	_	_	_	-	_	-	_		_	_	_	_	_	_	_	_	_	_	_	_		_	_	_	_	_	_	-	_		_	_	_		_	_	_		_	_				_	_	_	_		_	_	_	_	_	_

• Ellipse

_																																						_																									
Г	•		٠	•		•	•		ᄀᄃ	•	•		•	•	•				•	•		•	•	-	,	•	•				-						i i																		0				4	0		0	•
	•	•		•		•					•	•	- 4		•	•	•	11.		•	•			•										Γ.			r I	Г																46			0				0		
Т					11	•	•	•	11	•			. 1	•	•				•			•		۰.		•		Γ.	• -	•	۰.		1	1	<b>-</b>	1			•							7	-		15	1	1			11									
									Ш.				- 1							• -	•	٠.	Γ.	• -	•	٠.	۰T	Þ.	۰.	۰.	•-	ali	Γ.	~	a C	٠.			1	<b>—</b>	a C		11	77		-	70		11			2		11	1			× .				Ξ.	
	· .			· .	Ш.	. `			11		۰.	· ·	. 1		۰.	۰.		JI ~		۰.	Ξ.	. 1	<b>.</b>	۰.		• '	·	Γ.		•	۰.		~	-	۰.	× 7	1	Γ.	-	-		-	1		-	- 1		-	18		~		-	. 16		1	×	~		۰.			
Т			•	•	11	• .	۰.	÷.,	1.	•			· ]	Γ.	. •	. '	•	11_	Ξ.	. •	-	• .	Γ.		<b>`</b> ~	۰.	. –	L'	۰.	۰.		-1	۰.	~	~	٠.	<u> </u>	~		-	-	٠.	1	$\sim$		-	٠.	-	л.	-		1	•	11.	ч.	•		<b>۲</b> .	Ľ.	. `		۰.	
	•	•		•		•	•	•	11.		•	•	- 1		•	•	•	"		•	•		L '	• .	•	. •	• .	Γ.	. •	_ •	•	9						Ρ.		_	Ξ.		-11						45		-	-		. P					2.1	-	-		
L	•		•	•		•	•	•		•	•		•	۰	•		•	٩.	•	•		•	•							•																			10						0			0	4			•	

• Ellipse A

_									_						_																		_							_																				_													_							
	•	•		·	F۰		•	•	70		•	٠		٠	ī		٠		,	•		•	•	,			Ð		•		•		٩ſ		,		,					0		υ	_				-		-									R				L.		τL				1			1							
1.		•	•		I I	٠	•		· II	•	•		٠		4	٠		•	٠		•	•				•	L	٠		•		•							•		0		0		•	1	D	1	D		0		DE							D	1		(		(	Ш		E	0	E	(									Ĵ
	•	•		•	۰ ا		•	•	- 11			٠		•	.∥.∢		٠			4		•					•		٠		•						1		ι÷.		) _		) -	•	-		٢.	D	- (	D		D	1	m	6	Т	-			r	D)	1	) T	1	ĩ	(I		í –	E	(	E	í –	3							
1.		•	٠			٠	•		- II	•			٠		4	٠		•	٠		•		•		•		Ľ.,	•	_	•	Ē.,		1h	D		D		Ē		IГ		<u>۱</u>		г <b>т</b>		-1	D	- (	•	1	•	1		0					-	D	1	Ŷ	ſ	) –	ſ	ΣĿ		E		F	6	E	1							l
				•	۰.			•	- 11						Ш.		٠			4		•				. 1		-	•	Ξ,	•	٠,	١ſ		•		۰.		· -		۳.		۳.		- ۱		٢.	•	1	٦.	1	2	1							1	ì	1	ì.	ſ	ř	ŕI.			F	6	F	6	٦ŀ				1		17	
Ι.					L				. 11						1			•			•	٠.	•				Γ.	•	-	•	۰,	•		•		•		۰.		1E		-	•	-	a			-	•	1		1	N.	$\sim$			- 1		- 1	n	1	ì	1	ì.	$\sim$	М	$\sim$	-	6	-	6		4	-7		- 1		1		ł
				.	۱.				- 11	÷.,					11.			٠.		Л	Ξ.	•		. "		. 1		-	•	Ξ.	•	۰.		٠.	• "			-	-	1	~		~	-	-		2	•	-				7					-		1	Ň	1	Ŷ	1	Y	1	Ē	6	-	6	۰.	~	٦.		2				1	

• Ellipse B

_																																																			
	•	•		ī	•	•		,	• •	•	•	•	•	-	•		•		-		•			-												0			70									- <b>F</b>	( )	1	1
1.										- 4			•	114				el	•				)	•	•		۰ ا	)	)		•	•		-		F	•	6	1		Γ(	Γ(	<b>S</b> 10	6 -	1	1		1	1	1	
1 I							л.			. 11				4					۰.	۰ī -	•				- A			•					1				-		217					-	1	-		6.7	ć.	1	1
Ι.		. `		1 °	. `		1.		· . ·	· ]	÷.,		· · .	11.				1	<u> </u>	- N	-	. – L	· ·	<u>م</u>	<u> </u>			× .	<u> </u>	- 7		<u> </u>	~~			-	~				- /	-		1.	1.			1	1		1
11		•	•		•	• •	<u>۱</u>	•	•	- 11				. 11 -	· . ·	•	•	٦١	• .			·	·	• - '	•.	. <b>T</b> I	L		·	<b>7</b>		<u> </u>	Ο,	_	1	-	•		4.2	-			_/					<u> </u>	<u>,                                    </u>	1	
	•	•	•		•	•	11 1		• •	• 11	•	•	•	1	•	•	•				•		•					ዏ.		.9		-0	-		₽.		_		211								- 11			-	14
1.					•					- 41			•	- II 4				•	•				) (	•	•						•	•	•	•		F	•	Ø`			Γ.	Γ.	<b>7</b> 1	ίσ.	7	10		1	×	1	
1				1.			4 .			•				4							•		•	•	- <b>(</b>	•	•	•	•				~						917	T	c,	1		1	1		<b>1</b> 1		ć ,	1	1

• Ellipse B2

_												_																												_				_					_							
Г	•	•		ПГ	•	•	•	ᄀ	•	•			•	•	•			•	•	•		,	•	-			•		-					-				-												Ζ.			1	1		,
1				11				11.				- 41		•		•					•				•				•		•	- A	) i	•		- A	Γ.	o T			5	•	•	- 1	Γ(	<b>-</b>	65	1	11	11		1 1	1	1	1	1
				- II.				1				. 11		٠								•	•		- i		•		- <b>4</b>		<b>.</b>	•		- 4		•		-			1				6	1	- 1	-	12	1	10	1	11	1		÷.,
				11				Ш.										۰.	-	• ·	<b>_</b>		- a	• -	•	ها	- a	<u>، م</u>	•		-	<b>~</b> 4	<u>م</u>	•	16	-	× .	<u> </u>	$\sim$						20					1			1	1	1	1
				11				11	۰.	۰.	. T.,	. 11	. 1		• .	· .	<u>ا ا</u>	. *		۰.	٦١,		~	۰.			~	-	• -	. –	~~	~	-	•	12	-		•	-	16		<b>-</b>	~		<u></u>		_	-	4 -	/	1	1	Z		- L	<u></u>
	•	•	•	1	•	•	•	1	•			' 11	•	•	•	1		•	•	•		• .	•	. •			•		_			•		_	14	۰.		~	-											1	-		1			1
		•	•	11	•		• •	11		•	•	- 11	•	•	•	•	•			•	•	•		•	•			• . (	•	9				•	J	_		Ζ.		1.0		₽.		_					11	14			4		1	
L							•		٠						•			•					•						_			•		. (		•	•	- (				Γ.	E (				Γ.			1	1		11	1		,

• Elliptical 1

												_																																							
•	••••	•	•	•	•		• •		•	•	•	٩.	•	•	•		•	•	•		•					•	- 6											٩D										9	0		0
1	•	•	•	• I	•	•	•	- 11	•	•	•	11.	•	•	•	•	•	•	. (	•		•			179			•						ч				T	0	C			- (h			0	0				
1 •		•		.	•			- II.	•	•	•	4	•	•	•	, II.	•	•	•	•	- e	- T	<b>N</b> (	<b>b</b> 1	•	-	•					1	-11			11		٩r					ч	0	0			d 🗌			
1				•		•	•	-11	•		•	Ш.		•	•	•	۰.		, <sup>–</sup> (	• 1	• -	•	•	~e			•	•	167				-11	¥			Υ.	1		× ^	1		- 11		1						
1.				.	· .			. 11	•	•		Л.	· •	٠.	· .	, Th	• -	• -	•	∼⊣	<b>~</b> •	. Te		• 1	<b>L</b> .	<b>-</b>	<b>-</b> •		JE	•7	-	<b>`</b>	-16			$\sim$		11	м			<b>۲</b>	1				~	d T		۰.	
1 ·		· .	·	· .	1. 1				•	۰.	· · .	11.	Ξ.			. JI.	•	۰.		. 7	L -	<u>`</u>	<b>.</b>	•	Γ.	. –	~~	•	167	-	-	-	-11	$\sim$	•		-	٩Ŀ		<b>-</b> ^	~		-11	~	~		12	1	×.	. `	1
1	•		•	• 1	•	•	•	-11	•		•	111	• . •	• .	۰.	- TL	. •		· . '	• .	┍╸	•	. •	. •	L.,	·_`	•_	▝		_	_	~	-11	~		$\sim$	-			•		С.	. TP			ч.	~	. •		۰.	
•	• •	•	•	•	•		• •		•	•	•	٩.	•	•	•			•	•																			ЧĽ						$\circ$	$\odot$			<u>ا</u>			3

• Elliptical 2

_																																																	_													_											_					 _	
Г		•		•	•	Л	•		•	•		1	•		•		•	•		٠		•	٠		٠		•	•	•	•		•				٩.					•			. (	•	L.'	•	_ (								1	-						ι.									1	М					£	۰.
L	•		•					٠			٠	ı١		٠		۰.		1	٠		٠	•	· 1	٠		•			•	L	•			•						٠		•	Ľ	۰.	_ 1	•	L 1	۰.	. 1	D.		١.		)_		NE	e	1	•		۵.						I.					1	C I	1		1			a
L		•		•	•		٠		•	٠		1	•		٠		•	II۰		٠		•	•		٠	•	•	•	•	•		•	•		•		•		٠	, ,	٠		D	1	•	1	•	1	D.					•	)_	1		٠									Ľ	1		1			М		1		1	5	
L	•							٠			•	. ا		•		٠		1	٠		٠		.	•		•		•		L	•		•		•		)		• "	٠	_	•	170		-	•	1		-	Þ.	-0		10		1	١Г	D			٦.		٦		C					1				C I			1			5
L		•			•		٠		•			11	•		•		•	. ا		٠		•	•		٠		•	•		•		•		•		١ľ		•		, <sup>–</sup> (	٠			-	•	- 1	•	-			•	1		10	<u> </u>	1					٦		N.		1	7	P	1		1		٦.	М		1		1	5	
L	•							•			•	∥.		•		•		1	•		•		. 1	•		•				1 <sup>-</sup>	•	Ē (	b (	٠.		1	•		۰-		-		Г	•	7		-			Þ.	-	Ν.	1	<u>`</u>	0	٩Г	The second	Γ.			Ν.	٦		T					1			N				1			h
L		•							•			11	•		•		•	ш.		•		•			•				. 1	•	-,	•	٠.	•	٠.	1	٠.	ν.		. – I	•			٠,	•	_		٦,		20		1	<u> </u>	10	<u> </u>	1					٦	N				1	Þ	1		1	N	1	М		1		1	5	i.

• Elliptical P

				_									_																																	_					_								
	•	•	•		•	•	•	ᄀ	•	,	•	•	I۰	•		•		•	•		•	•	•			P	-		•							- 1										١D			0				0			0	0	•	ł
1.			•		•				•	•			4	•	٠	•	11 •		•	•	- ell		•	•		. 18							רו			- 11															0	0							d
					•	•	•	- 11	•		•	•	. اا	•		•					. II	•	•	- e	•	- elli	Ĩ.	<u>، ت</u>	• `	۰.	٦	•	'n.													ЪĿ		1				1							
Ι.																		· · ·	• -	•	· .	Ĩ.,	<u>،</u>	•	۰.	. TI		•	٠.	ν.	الم			•	``			٠.	•			10	-	1	٠.	71			<u>~</u> г	$\sim$		1	~	•		۰.			ł
1			÷ .	Ш.	. `	. `			۰.	÷.,	. '		11 .		-	. · ·	٦Ľ		۰.	Ξ.	. 1		·	۰.		പ്	· .	. –	~ "	•	. —		~	<b>~</b>		_	`	_	- 7	÷.	-		1	•	-	16	Ч-	~	-		Τ.				×.	. `			
	•	•	•	11	•	•	•	11	•		•	•	• ۱	•		•	11	•	•		• 11	•	•	. •	• .	- <b>T</b> I		·_`	•_	. •	<u>_</u>	Β.	. =		<u> </u>	-		_	۰.		_1	-	٩.				~		<b>_</b> .	$\sim$		1	<u> </u>	~	~ 1				
1.		•	•		•	•	•		•	•	•	•	11	•	•	•	•		•	•	୍ୟା		۰.	•	. •	' . II	•_	•	_ •	•	9		U			_					ч.						Ο.				Φ.	$\cup$		-					ł
	•	٠	•	_	•	•	•	⊐∟	•		•	•	•	•		•		•	•		•		٠					-			_					- Y										ЧD							0				0	•	

• Elliptical Q1

$\cdot$		
· · ·	 	
∥		
1		

• Elliptical Q2

-																_							_	_	_			_	_		_		_		_	_	 	_	_	_	_	_	_	_	_	_	_			_		 _			_	_
Т	•		•	•	· II	•	•	•	- 11	•		•	•		•	۰.	•	41	•	•	•	• 11	•	•	•		91	•			•			•			•	•	•	- 11	<u> </u>	Δ.	-		MN.		•					. N	<u>ر</u> ۲	<b>N</b>	× .	1
Т					- 11				、 II					11						•	•				•			<u> </u>				1.			~	• 1		-	1		•				IN.									1	~	
Т					- 11				· II					11				11.1		•	•	- TI			• .	-	- 12		•_	•		- L '	•_		_	_			<u> </u>	-			Ŀ.`	<u> </u>					. 11	· ·			~			
			•	•	· II	•	•	•	- 11	•		•	•			•	•	41	•	•	•	• II	•	•			•	•						•	•		•	•	•		• •		-		NIN.	A 7	•	•					. ``		× .	
Т					- 11				. 11					л								الم	- <b>•</b>		•	•	76	<u>، -</u> ،	•	<b>.</b>	٠.	10	<u> </u>	-	~			-			•	-			١Þ.,						. ``			$\mathbf{i}$	~	
	•	•		•	- 11			•	• II		•		•	11				. 11. 1		•	•	- 11			•			•	•	•			•			D.			<b>_</b> `						1			~					~			< .
Т			•	•	. 11	•	•	•	- 41	•		•	•	11.5			•	41	•	•	•	• II	•			•		•					- 0			۰.		•		7		•	-									_ N	. ``		× .	
					- 11				- 11									Ш.	-			· .II		Ξ.		·	ᇻь	_		· · ·	⁻_			- C	∽_		- N	<b>-</b>			-	<b>_</b>	<u> </u>				-				. ``		$\sim$	$\sim$	$\sim$	-
1	•	•		•	- 11			•	۰ II	•	•		•	11	•	•	•	119	•	•	•	ા								•							-			-	-		• •		48				- 11	- A				~		
Т					. 11				- 11									Л				. 11	•		÷.	۰.	- II -	•	- <b>b</b>	. –	•		<b>~</b> e			<b>-</b>	2		•	-														. ``	1	$\sim$
L			•				•	•				•				<u> </u>	<u> </u>	-				•																		_									_			. '			~	-

• Euclidian

_																																																			_								
	•	•		٠	•	•			1	٠	•	•	ТГ	•	٠	•	•		٠	•		•	•	-	•	•					•	שר			•	•										Q D							$\circ$				•	•	•
1.		•	•			•	•	•	۰ اا		•	•	41	•		•	•	٠			٠			•	٠				•				•		- 6										T						0								
	•			•	•	•			11	•				•	٠	•			•			•	•			•		- <b>e</b>	- C	•	•		~		•					11	-11		۲r	м		ΠD	1	1				2	0				•	•	
1.		•				•		•	١.				41			•	•						Γ.	• -		٠.	- TI	•	•	•	٠.		•	`	-			1				Y			~	11			1	~	<b>d</b>			1					
					Ι.				1		۰.		11			· .		-	-	· .	-	- 1	<u>م</u> ا	-	Ξ.	• "	<b>_</b>	<b>~</b> _	· · ·		•	16	٠.	<u> </u>	•	<b>_</b>			- 7		-11		~	-		46	×.	~			112			× .		Ľ.,			
				·	<u>۱</u>	. 1			Ш.	•	. *	. •	1	• .	•	. •	. 1		۰.	. *		• _	Γ.		-	۰.	. Т	L	<u>'</u>	-	۰.	JE	<b>_</b> '	-	۰.	5			۰.			$\sim$	<u>.</u>		-	٩Ľ			ч,	~				<u> </u>			۰.	. ×	
1		•	•			•	•	•	""		•	•	11	•		•	•	•	. '	• .	•	. "	L.'	•		. <b>-</b>	' _				_	16	Ψ,		~	·	Π.		_	~	- 11	~		-	~	45	-	$\sim$			Ψ.				-		. •		
	•	•		٠	•	•				٠	•	•		•	٠	•			•	•		•	•		1		•						_													ЦU											•	•	•

• Inverted Double



• Inverted Double Dot

		 					-		_	-			_	-		-	-	-	-		100					_	100			-	-	_											1.00						100							-						-	-		_	 
 				II.					• II.		•		•		в.	•	• •	• •	۰.	•	H-	• •		۰.	•	•	11-4	-				Γ.				- 11	ne i	ЪC	ЪT	i n		<b>C</b> 1	1 nu	~		~	ЪП	-	10.0	-	-	•		~							- 1		•	•		
 	•	 		ш.					- 41		•				41	٠.	•	•	•	• •			•				lle i	•		-	- 1	67				m	n	0	<b>m</b>			201		ì		Ο.				-		-			10											
 				II.					· II.						н.						H .			· •	•	•	11-1	-				-	_	-	_	- 11		10				0	in.	~																						
 		 		ш.					- 41		•				41	٠.	•	•	•	• •			•		•		lle'		• •	-	- • 1	67	•			<b>m</b> 11	n	0	<b>C</b> .		20	201	11-	$\sim$	-	$\sim$	~			-	-	Π.			10				-							
 				п.					. 11						н.		• •		•	•	н.	• •		· •		•	11-1	-				<b>-</b>				- 11		207	20			~	10.	~				-		_			_	-							· I					 
 		 		ш.					्या	•	• •		٠.		41.	٠.	•	•	•	• •		· • ·	•	•	• •		lle'		• •		- 1	•	• " •			- 11		C I	C a	<b>.</b>		10	11-1	YO	-	$\sim$	~			-	-	~			10				-					۰.		
 				п.					. 11						п. –		• •		· •	· • ·	H-	• •		· •	•	•	11-4	 -				<b>-</b>	-			- 11		200	20	s na s		~	in-	~				-		_			_	-		_										
 		 		ш.		•			- 41	•	• •	· •	•		н.	٠.	· • ·	•	•	• •		· • ·	•	•			lle'		• •		- 1	67		-		-11		<b>m</b> .	<b>C 1</b>	<b>~</b> ~		201	11-		-	Ο.	<b>~</b>			-	~	~	- 1	•	10				-							
 				н. 1					. 11	· •	÷ .				п.	•	• •	· .	· •	· • ·	п	• •				•	11-4	 -				<b>-</b>		-		- 11	<b></b>	207	ю.	w n		<b>C</b> 1	in-	~		~		-	10.0	_	-		_	~					•		- 1					 
 	1.1	 	· . ·	ш.	÷.,				ं अग		• •	· .	· • ·		и.	٠.	· • ·	•••	•	• •		· • ·	•	•		· .	lle'		• •		- 1	67				- 11	co.	<b>C</b> 1	<b>C</b>	<b>~</b> ~	- 20	201	11-	× -	-	<u> </u>	~			-	-	~		~	10			-	-	-						
 				II. 1	•			•	· II.		٠.	•	• •	· •	п.	•	• •	• •	٠.	· • ·	H	• •	· •	· •	•	•	11-4					Γ.	-		Ξ.	- 11	n e	ЪC	ЪT			<u> </u>	10.	~				-		-		•	-	~	11						• I			•		
 · · · ·	· · ·	 · · · ·	· . ·	п.	۰.			· ` •	: II		۰° .		· . ·		и.	۰.	· • ·	•••	• *	٠° .	н.	· • ·	•			· • •	lle'		• •	-	1	67		_		- 11	ю.	ce e	с×.	сw.	- 20	201	11-	× 7	-	~	<b>~</b> ~			-	-	~	~	~	16			-	-	-						
		 		н. т	• `	· .			. 11	· •	٠.		• •	· •	п.	•	• •	• •	٠.	· • ·	II	• •	· •	· •	•	•	11-,					Γ.	-			- 11	ne i	207	ю.	-	- 22	~	10-	~		~	× 1	-		~	~	•	-	~	11		- 1				•					

• Inverted Ellipse A



• Inverted Ellipse B



• Inverted Ellipse B2

· · · · · · · · · · · · · · · · · · ·		
• • • • • • • • • •		

• Inverted Round

				ĸ	ĸ	ĸ	1			×	-1				×	X	1	ĸ	P C	X	N.								Т					ж																		•	•		
1						×	1				_ II					×	×	×		r i	<b>1</b>	¥.	1	Πì	- C	¥.	Y	- 1	റ		Y	Y		$\mathbf{n}$		Y			T	T		ΠĽ	11	7			12	`	$\sim$		- 4				
1.							Ш.,		-		- 11		-				· · ·	¥	J٢	` <b>∀</b> ´	<u> </u>	<u>с</u> ,	-7	1	∽	$\sim$	$\sim$	чr	$\sim$	~	$\frown$			$\sim$	~	$\frown$	-11	$\sim$			~	71		-	-	~~	45		-	-					-
1.		~					11.1		°	÷.	- 11	. 1	۰.	÷.,	11-				ᇺ	<u>~</u>	_~	<i>`</i> ت'	-ي		~	$\sim$	$\sim$	٦L	~	_		$\sim$		~		$\smile$			-	$\sim$		41	-	~		-	NĽ.	<u> </u>			- 11	× .			
- I			× ۱				11				' II	•			11					U.J	۹	. Т.		עו	۰.	Α.	ᅳ		U	┺.	ᄎ	~			٩.,	А			~			٩L		_		_	46		<u> </u>	<u> </u>	- 4				
		*		×	×	×	11 *				-11				≍	<b>X</b>	1	×	쎼	Ξ.	- 1	)			_ <b>_</b>		ι.		黒				K.	ж			-41				П.	л	$\odot$				(P)					•	•		
					×	×	11	ж.			- II				41	X	×	×		()		×		11	1	X.	X		()	Υ.	X.			C 1	× .	x						Π					16							•	
					×		11 .				- 41				ll z	×	1	×	ulf -	Υ.	_` <b>`</b>	C i	<b>-</b> 1			$\sim$	$\sim$	чF	Y		$\cap$	$\sim$		Y		$\cap$	-	м			m	1			11	<b>٦</b>	db.			1					

• Line



• Line 90



• Line X



• Line Y



Rhomboid

	•	•					·Π			┱	•																_										1								Πh								1.1
1			• II				- 41			Ш.				- 41				- Ih				•						161		$\Box$						1					îП				16				- 1				
1.			- 11				. II			4				. 11					- P	- T	•	•		- 1			-	IC.	• 7	ΞŤ	- n			5	~	$\frown$		w			71				16								
			, II				- 11			Ш.					۰.	· -	•	٦h	. 7			<u>-</u>		1	-	· <b>-</b> I	- 7	њ.	- n	-	•7	'm l		$\frown$		5	71		•		10	-		۳n	11				- 1		• -		
Т.			- 11				. II		÷.,	1				. 11	•		۰.	 -II			- 7	۰.	ᇻ			1		IC 1	• 7	12	-		<b>1</b> 77		-	-		10			-11				16		-	-			۰.		-
	· .		. II	. 7			· II	٦.	 	11.			_ `	- л	Ξ.		- 1	 ٦.		-	۰.	. –	ᅬᄃ			Ξ.	- 7	16.	-	-	• 7	'ad		-		۳.	-16		- 7	-	10	-	-	-	11	-		-	- 1				
	•		`	•	•	•	- 11					•		- 11	. 1	•	۰.	 _11'	•			<b>'</b>	ᇺ				•	JP.		_	۰.	-		_		-	!!	_	۰.						1.	-	_			_		_	-
	•	•				•	· .			1	•				•		_																												UU.								

• Round

_																		_																			_				 																
Г	•	•		٠	•	•		•	1	•	•	•		•	•	-	•	4	•	•	•		•	•			9							•					•	_		0				$\cup$											
		•	•		I 1	•	•	•	۰ اا	•	•	•	- 41	•	•	٠	•			•	•	•			•				•		•		•			•		-	) (	•	r 🗲	) (	•	•	- 1		-		-	110							ŝ
	•			•	١.	•		•	11	•			· II	•				4	•			• IIe	•	٠					- <b>(</b>	D.	•	-	) <sup>–</sup> (	•	•	-		•		-	D-	$\sim$	-		٦ľ	$\cap$	~	$\mathbf{T}$		íľ.		11		1			
		•			L	•							- 41			•				•	•	al I	۰.	۰ī (	•			١Ť.,	•	- <b>-</b>	<u>، ا</u>	٦		-	•	•	15	~	1	•	10	Y	~	$\sim$	-11	$\mathbf{Y}$	-	$\frown$	-	٦ŀr	-		1	1			f
				.	١.				1				. 11				· · ·			· .	٠.	. 1	• "		۰.	• -		-	٠.	•~	•	7		•	<b>`</b> _`	~_		•	$\sim$	~			-	$\mathbf{\nabla}$	- 11	$\sim$	~	$\mathbf{\nabla}$		<b>/</b> ľ`		1	~.				
				·	1.	. 7		· .	Ш.	۰.	. *	. *	. II			. '		11_	Ξ.	. *		' II'	• -		- "	•	ᇻ		-`	-		ᆈᄃ	<u>`</u>	۰.	. –	<u> </u>	1Ь`	•_	<u> </u>	~~		$\sim$	~	~	-1	$\sim$	~	$\sim$	-	NI.	-	<u> </u>	ີ.	1		 	i,
		•	•		I 1	•	•	•	<b>.</b> ا		· .	•	- 11		•	•	•			•	•	- <b>1</b> 1	. •	' . '	• .	. •			•	_ •	· . '	- 1		~		•.		_	~	•_		_		$\cup$	- U.	$\sim$		$\cup$	-	ж			Щ.			-	ł
	•	•		٠	•	•		•	1	•	•	•	· _	٠	•		•	4	•	•																				-					21												

• Square

• •	• •	• •		• • •	• • • •			<b>X X X X X X X</b>	<b>x x x x x </b>
	- II •	• •		4 • • •			`		$\mathbf{X} \times \mathbf{X} \times \mathbf{X} \times \mathbf{X} \times \mathbf{X}$
	·    •	• •				<b>● ` ● ` ● ` ●</b> ` ●	▖`♠`♠`Ib`♠` <b>≜`</b> ≜I' <b>≜</b> `4		$\mathbf{X} \times \mathbf{X} \times \mathbf{X} \times \mathbf{X} \times \mathbf{X}$
1							´▲`▲`▲ ` <b>▲</b> `▲`▲`  <b>\</b> ` <b>&amp;</b> `		
1									
1	́Ш".	• •	∥. · . · . `	▼▼.	1				
$ \cdot \cdot \cdot$	1 1	• •		1					<b>A // A A A</b> // A A .

• Square 1



• Square 2

• **HCS**\*

# \* A special note on HCS

**Harlequin Chain Screening** uses an elongated dot shape to create better reproductions of certain images such as computer-generated graphics. Instead of the conventional symmetrical dot shape, HCS uses a long elliptical dot to produce a chain-like structure. The elliptical dot creates particularly smooth flat tints and vignettes even when working at relatively low screen rulings.

- WARHWARKAA	WARDON MARINE	000000000000000000000000000000000000000	YOHDIOKOHDIOKOHD	MAANAMAANAA MAA

# Why is the Yellow screen ruling higher than the other process colors Cyan, Magenta and Black

This is all due to the limited screen angles that are available for simulating halftones. RIP technologies typically use screen angles in 30° increments for CMK, which results in a nice triangular pattern. Yellow is in between, in a 15°. In order to match the pattern of the other separations, its ruling needs to be bigger:

YELLOW HAS SAME RULING AS CMK

When yellow has the same ruling as CMK:

When yellow has a higher ruling than CMK:



More specifically, to match the pattern of the other separations, the Yellow screen ruling should be (2/sqrt(3)) \* the ruling of CMK, which is around 15,4%. Taking into account a fault tolerance explains the 14% difference in screen rulings.

# **HDS screens**

Harlequin Dispersed Screening is a second- generation stochastic screening that resolves many of the quality issues of conventional screening, such as moiré, graininess, dot gain, dot loss, mid- tone jump, banding...

It creates reproductions with increased sharpness compared to conventional screening methods and provides freedom from both cross screen and object moirés.

# **Screen Structure**

Although the screen structure of HDS contains a random component, there is an underlying ordered, though dispersed, structure. Dispersed screens produce structures with a high perimeter to area ratio. This allows for sharper reproduction, even at relatively low scanner resolution. HDS is ideal for reproducing subjects with moiré issues and where detailed reproduction is required.

#### **Speed and Performance**

While most FM and stochastic screening methods typically reduce the speed and performance of the RIP, HDS can be implemented with no performance and speed penalty.

#### **Higher Quality Images**

HDS eliminates the grainy highlight effect experienced with some FM screening methods that are particularly apparent in smooth areas of contone pictures, flat tints, and vignettes. These are the result of purely random screening methods that cluster dots into odd sized and shaped structures. HDS uses a dispersed but still clustered technique to produce a screen that has the benefits of both stochastic and conventional screens. The clustering improves dot gain characteristics, plate lifetime on the press, and ability to produce a representative proof while the dispersed component eliminates moiré and improves image quality.

Since individual spots do not randomly cluster into structures with odd sizes and shapes, graininess is reduced compared to other FM screening methods. At the same time, the underlying structures are so fine in scale that the benefits of increased sharpness are maintained. Though there is no regular and periodic structure, there is a certain regularity of form and scale. If you examine HDS tints closely, you will notice that the tints have a certain homogeneous structure that changes smoothly up and down the gray scale. This consistency in structure at a particular gray level or dot value eliminates many of the problems of controlling printing conditions associated with purely stochastic screening methods that contain structures of many different sizes at the same dot value.

#### Family of screens

HDS provides a family of screens, tailored for different types of reproduction and different output device capabilities. HDS includes standard four colorant screens for use with the standard CMYK process colors as well as variants developed for hexachrome and photo ink devices. HDS is also available in five varieties of dot structures or settings making it compatible with many different types of output device: • HDS Super Fine: this screening is a solution for low resolution output devices.



• **HDS Fine**: this screening produces extremely smooth grain-free results and has the finest structure of any HDS setting. It is intended for use with high quality output devices capable of recording single isolated laser spots on separations. HDS Fine works best for high quality commercial printing environments with tight prepress and printing controls.



• HDS Medium: this screening offers a slightly larger dot structure in the midtone range. It can be used in intermediate commercial and publications printing, either sheet fed or web.



• HDS Coarse: this screening makes use of somewhat larger structures to increase printability and is best selected on presses which cannot hold very fine detail such as in high speed web printing.



Note: The HDS Coarse screening has a double version as well, for even coarser results.

**HDS Super Coarse**: this screening uses a larger structure to increase printability and the capability to retain highlighted areas of an image. It is suitable for use on mid- to low-range image setters or plate setters with resolutions between 1000 and 1600 dpi.



=

Note: The HDS Super Coarse screening has a double version as well, for even coarser results.

#### Which dot to use?

The selection of which type of HDS dot to apply depends on the physical characteristics of the output device, the type of ink and paper. The physical size of the HDS dot created with each of these options will vary both with the device laser spot size and the resolution at which it is driven.

In this table you can find the theoretical spot sizes in microns for the different HDS models and different output resolutions for 20% HDS screen:

Image setter/ plate setter resolutions dpi	1016	1200	1800	2400	3600
HDS Super Fine 1x1 pixel	25	21	14	11	7
HDS Fine 2x1 pixels	35	30	20	15	10

Image setter/ plate setter resolutions dpi	1016	1200	1800	2400	3600
HDS Medium 2x2 pixels	50	42	28	21	14
HDS Coarse 2x3 pixels	61	51	34	26	17
HDS Coarse Double 5x5 pixels	122	102	68	52	34
HDS Super Coarse 4x4 pixels	100	84	56	42	28
HDS Super Coarse Double 4x4 pixels	200	128	112	84	56



**Note:** HDS is not limited to the output resolutions listed.



**Note:** In case of HDS screening, rulings and angles are not relevant.

# HXM screens

Harlequin Cross Modulated screening is a hybrid AM/FM screening that allows higher screen rulings than provided by AM screens alone.

In conventional screening technologies, the level of detail that can be reproduced by a given system depends on the size of the smallest printable dot possible on paper. Higher screen rulings produce very small structures in the highlights and shadows. These are inclined to be lost, which results in loss of detail.

HXM screening combines the advantages of both AM and FM screening and is therefore able to limit the loss of small printed structures and detail at high line screen rulings by controlling the dot size in the highlight and shadow. It limits the smallest structure produced to be that which can reliably be printed on the target system. Once this point is reached within a given HXM screen, structures are removed completely rather than continuing to be reduced in size to create the required tonal range. Therefore, it does not disappear in the light areas or merge in the dark areas, which improves the print quality with smoother flesh tones, flat tints and gradient transitions.

The following HXM screens are supported:

- Flexo Hybrid
- Offset Round
- Offset Traditional

#### **HXM screening sets**

These tables provide an overview of the supported HXM screening sets.

**Important:** Because HXM is a pre-generated tile based screening, it is not recommended to mix it with function based dotshapes as the RIP will not take the HXM ruling into consideration when calculating the actual angles and ruling for the function based dotshapes.

#### **FLEXO**

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
2000	133	25
2000	133	38

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
2000	133	51
2000	150	25
2000	150	38
2000	150	51
2000	175	25
2000	175	38
2000	175	51
2400	100	11
2400	100	21
2400	100	32
2400	100	53
2400	120	11
2400	120	21
2400	120	32
2400	120	53
2400	133	11
2400	133	21
2400	133	32
2400	133	53
2400	150	11
2400	150	21
2400	150	32
2400	150	53
2400	175	11
2400	175	21
2400	175	32
2400	175	53
2400	200	11
2400	200	21
2400	200	32
2400	200	53
2400	280	11
2400	280	21
2400	280	32

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
2400	280	53
2540	55	60
2540	55	50
2540	65	60
2540	65	50
2540	75	60
2540	75	50
2540	85	60
2540	85	50
2540	100	10
2540	100	20
2540	100	30
2540	100	50
2540	120	10
2540	120	20
2540	120	30
2540	120	50
2540	133	10
2540	133	20
2540	133	30
2540	133	50
2540	150	10
2540	150	20
2540	150	30
2540	150	50
2540	175	10
2540	175	20
2540	175	30
2540	175	50
2540	200	10
2540	200	20
2540	200	30
2540	200	50
4000	100	13

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
4000	100	19
4000	100	32
4000	100	6
4000	120	13
4000	120	19
4000	120	32
4000	120	6
4000	133	13
4000	133	19
4000	133	32
4000	133	51
4000	133	6
4000	133	63
4000	133	76
4000	150	13
4000	150	19
4000	150	32
4000	150	51
4000	150	6
4000	150	63
4000	150	76
4000	175	13
4000	175	19
4000	175	32
4000	175	6
4000	200	13
4000	200	19
4000	200	32
4000	200	6
4800	100	11
4800	100	16
4800	100	26
4800	100	5
4800	120	11

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
4800	120	16
4800	120	26
4800	120	5
4800	133	11
4800	133	16
4800	133	26
4800	133	5
4800	150	11
4800	150	16
4800	150	26
4800	150	5
4800	175	11
4800	175	16
4800	175	26
4800	175	5
4800	200	11
4800	200	16
4800	200	26
4800	200	5
5080	55	60
5080	55	50
5080	65	60
5080	65	50
5080	75	60
5080	75	50
5080	85	60
5080	85	50
5080	100	10
5080	100	20
5080	100	30
5080	100	40
5080	100	50
5080	100	60
5080	120	10

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
5080	120	20
5080	120	30
5080	120	40
5080	120	50
5080	120	60
5080	133	10
5080	133	20
5080	133	30
5080	133	40
5080	133	50
5080	133	60
5080	150	10
5080	150	20
5080	150	30
5080	150	40
5080	150	50
5080	150	60
5080	150	70
5080	175	10
5080	175	20
5080	175	30
5080	175	40
5080	175	50
5080	175	60
5080	175	70
5080	200	10
5080	200	20
5080	200	30
5080	200	40
5080	200	50
5080	200	60
5080	200	70

# **ROUND OFFSET**

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
1200	75	11
1200	75	21
1200	75	32
1200	75	42
1200	85	11
1200	85	21
1200	85	32
1200	85	42
1200	95	11
1200	95	21
1200	95	32
1200	95	42
1200	10	511
1200	10	521
1200	10	532
1200	10	542
1200	12	11
1200	12	21
1200	12	32
1200	12	42
2400	13	311
2400	13	321
2400	13	332
2400	13	342
2400	15	11
2400	15	21
2400	15	32
2400	15	42

# TRADITIONAL OFFSET

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
1200	105	11
1200	105	21
1200	105	32

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
1200	105	42
1200	120	11
1200	120	21
1200	120	32
1200	120	42
1270	110	11
1270	110	21
1270	110	32
1270	110	42
1270	127	11
1270	127	21
1270	127	32
1270	127	42
2400	133	11
2400	133	21
2400	133	32
2400	133	42
2400	150	11
2400	150	21
2400	150	32
2400	150	42
2400	210	11
2400	210	21
2400	210	32
2400	210	42
2400	240	11
2400	240	21
2400	240	32
2400	240	42
2540	180	11
2540	180	21
2540	180	32
2540	180	42
2540	200	11

Resolution (DPI)	Ruling (LPI)	Minimum Dot (µ)
2540	200	21
2540	200	32
2540	200	42
2540	220	11
2540	220	21
2540	220	32
2540	220	42

# How to define the HXM minimum spot size (in $\mu$ )

You can calculate the equivalent percentage this way:

# Example

L = 150 LPI

 $R_0 = 2400 \ DPI$ 

Min dot  $50~\mu$ 

1. Calculate how much 1 pixel is:

 $1 \text{ inch} = 25.4 \text{ mm} = 25400 \mu$ 

 $(25400 \ \mu)/(2400 \ DPI) = 10.5 \ \mu = 10 \ \mu$ 

2. Check the size of the minimum dot = 50  $\mu$ 

One pixel is 10  $\mu$  so this means that this fits 5 times in the minimum dot.

| 10 µ |
|------|------|------|------|------|
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |
|      |      |      |      |      |

Which results in this raster to build up one dot. In this case it is a grid of 5 x 5 = 25 pixels.

 $(2400 \text{ DPI})/(150 \text{ LPI}) = 16^2 = 256$ 

(25 pixels)/256 = 0.097 = 9.7 = 10%

So a minimum dot of 50 µ at 2400 DPI and 150 LPI equals 10%.



Note: If the exact minimum dot is not available in the supported HXM screening sets, the next (bigger) minimum dot is used.

# Standalone RIP

The CLOUDFLOW standalone RIP option contains the following windows:

# Jobs

In the **JOBS** tab of the standalone RIP you can:

- Have an overview of all jobs that are currently running or have been running.
- Start a new RIP job.
- Have an overview of the finished job.

#### Job overview

The job overview provides you with an overview of jobs that are currently running and have been running. All this information is stored in MongoDB and can be called with the search tool.

You can search the job overview using various filters:

- The job name.
- The job status (Error, Hold, Running, Finished).
- The job creation/modification time (Modified Last 24-hours, Created Today, Modified Last Hour, Created Between).
- The flow (in case of a standalone CLOUDFLOW RIP you will only have this one available).
- The user (yourself or all other users).

When you select one of the jobs, you can:

- See a complete **overview of the job** in the right panel in case **see is selected** on the top right.
- See the **logs** time line of the RIP job in the right panel in case **I** is selected on the top right. The logs will give you detailed information on how a RIP job has been processed.
- Perform a specific action on the RIP job:
  - **Reload** a job (<sup>C</sup>). When rerunning a job again you'll need to submit the file(s) again.
  - **Stop** a job that is currently running  $(\mathbf{x})$ .
  - Delete a job ( 🔲 ). If you do this you will also delete the output files on your file server. A dialog window will ask you if you are sure about this action.
    - =
- Note: You cannot remove a job with status Hold.
  - Note: You can delete multiple RIP jobs at the same time by selecting them and pressing the Delete = key on your keyboard.

#### Start a RIP job

To start a new RIP job, follow these steps:

- 1. Select Create New... in the left panel.
- 2. Browse a file, or drag a file in the Upload Window.
- 3. Select a RIP preset. See Presets on page 350 for more information on the settings. You can also tweak a preset by first selecting this preset and then selecting Custom.
- 4. Select Submit.

### Overview of the finished job

When the job is finished, you will see a resume with information about how the job is ripped:

- Start: the input of the workflow behind the RIP.
- Current: shows the current node where the file is at the moment.
- State: error or normal.
- Finished: yes or no.

# Data

Here you can find a resume of the RIP settings used on the job. See Presets on page 350 for an overview of all the RIP settings.

# Files

In this section you can select the links to view the ripped files with PROOFSCOPE.

# Presets

In the **PRESETS** tab of the standalone RIP you can manage RIP presets.

You can:

- Create a new RIP preset.
- **Remove** a RIP preset.
- **Download** a RIP preset.
- Upload a RIP preset.

# **Create a RIP preset**

To create a RIP preset, follow these steps:

- 1. Select Create.
- 2. Give the preset a name.
- **3.** If needed, copy the settings from an existing preset.
- 4. Select Next to define the details of the preset by category.
- **Note:** You can select the eye icon on the top right of each preset (except **Separations**) to show or hide the presets in the Jobs overview or KIOSK.

#### Output

# **Output Type**

- **Output**: here you can define what kind of file the RIP has to produce. Options:
  - Composite: if you select this option, the RIP will generate a composite file
  - Separated: if you select this option, the RIP will generate separated files without a screen
  - Separated Halftone: if you select this option, the RIP will generate separated files with screen
- Bits per channel: here you can define the number of bits per separation. For example:
  - 1-bit tiffs =  $2^1$  gray levels = 2 = white (0) or black (1)
  - **2-bit tiffs** =  $2^2$  gray levels = 4
  - 4-bit tiffs =  $2^4$  gray levels = 16
  - 8-bit tiffs =  $2^8$  gray levels = 256
  - **10-bit tiffs** =  $2^{10}$  gray levels = 1.024
  - **16-bit tiffs** =  $2^{16}$  gray levels = 65.536

# Resolution

- Vertical: here you can define the vertical resolution of the output file.
- Horizontal: here you can define the horizontal resolution of the output file.

# **Output Properties**

- Pages: here you can define the pages that you want to output. You can select all pages or a range of pages.
- Rotation: here you can rotate the file for output.
- **Page Box**: here you can select the page box for output.
- Horizontal Distortion: here you can define the horizontal distortion.
- Vertical Distortion: here you can define the vertical *distortion*.
- Seamless screening: if you select this checkbox, the screening in the file is horizontally continuous between the jobs, from left to right. This means that, if you would put the beginning and the end of the output file together in a horizontal direction, there would not be any gaps or broken dots. Seamless screening is especially used when making flexo cylinders, for example for wall paper.
- Negative Output: if you select this checkbox, the output of the file will be negative.
- Mirror Output: if you select this checkbox, the output of the file will be mirrored.
- Enable Harlequin Precision Screening: if you select this checkbox, Harlequin Precision Screening (HPS) is enabled.

HPS is a color screening technology that ensures high-quality reproduction with any screening option in the RIP (defined using a spot function). It allows you to select any screen frequency and to use the usual CMYK screen angles of 0°, 15°, 45°, and 75° (plus multiples of 90°). To reduce moiré patterning, HPS uses an adaptive screening technique that can adjust each halftone dot so that it is placed within one half pixel of its ideal location. HPS also ensure rosettes are always hole centered, and it will generate extra gray levels, allowing the use of higher screen frequencies than the resolution would normally allow.

Enabling HPS screening sometimes results in another ruling or angle than requested. This is a consequence of the Precision Screening, which prevents rosettes from shifting. The ruling and angle are always optimal as HPS sets out to achieve the best possible match with the parameters the user defined. When you select the **Favour Angles** option, the algorithm favours angles to best match your parameters. When you select the **Favour Ruling** option, the algorithm favours ruling to best match your parameters.

# **TIFF Format**

- Strip format: here you can select if the output file should contain one or multiple strips.
- Compression: here you can select the compression of the output file.
- Anti-aliasing: here you can select the anti-aliasing type.

**Note:** Anti-aliasing is only possible in case of with 8-bit TIFF files.

- Bit order:
  - Reverse bit order: if you select this checkbox, the TIFF file will be output with reverse bit order.
  - **Pad to 32-bit alignment**: if you select this checkbox, each line of the TIFF file data will end on a multiple of 32 bits. This is an efficiency setting, for monochrome output only, that may make the file faster to read in some applications.

#### **Color Management**

In Color Management you can apply a color profile to the job.

#### **Input Profiles**

- Input RGB Profile: here you can select the input RGB profile.
- Input CMYK Profile: here you can select the input CMYK profile.

# **Output Profiles**

- Output Profile: here you can select the output profile.
  - **Ignore ICC tagged objects**: if you select this checkbox, objects tagged with an ICC profile will be ignored when applying a profile to the job.

# Post Process

- Wait on prerender: if you select this checkbox, the rendering of the file tiles for viewing in PROOFSCOPE is performed when the job is still running. In that case, the job will run a little bit longer but you can view it immediately in high quality.
- **Copy output files to external location**: if you select this checkbox, you can define an external location where the RIP needs to output the files (for example the station of the exposure machine).

Example paths:

- file:///Users/Name/Documents/Cloudflow/Rip\_Copy/
- ' 🥙 file:///D:/Program%20Files/Documents/Cloudflow/Rip Copy/
  - **Note:** You can only use local paths. Using network mapped drives is not possible.
- Copy CIP3 files to external location: if you select this checkbox, you can define an external location where the RIP needs to output the *CIP3* file.
- Copy with technical colors: if you select this checkbox, the output file will contain the technical colors.
- Ask operator for decision before copy: if you select this checkbox, the RIP will ask the operator to view the file when it has been ripped. The operator can decide if the file is OK or not. If the operator decides the file is OK, the **Post Process** options will be processed and the job will be completed.
- **Note:** You can always download or upload presets. This is very handy in case of sharing or backups.

# Separations

# **Default Screening Settings**

- Job Metadata
  - **Retrieve halftone settings from the Job**: if you select this checkbox, the metadata of the file will be used. If there are already screens defined inside the file, the RIP will take these to calculate.
- **Default ruling**: here you can define the ruling.
- **Default dotshape**: here you can define the dotshape.
- HXM Min Dot Size: here you can define the minimum dot size (in μ) in case you are using HXM screening. See HXM screening sets on page 340 for more information on how to calculate the minimum dot size and the supported HXM resolutions.
- **Rescale images**: in some cases it is useful to upscale all images in a file before ripping to improve the occurrence of broken dots by adding extra interpolated pixels in between. Harsh pixel transition is smoothened out with

cleaner dots as a result. The drawback is that this technique will lead to a performance hit and a slight blurring of the image data. Options:

• None: if you select this option, no image rescaling is done, each pixel is screened in itself.



• Automatic: if you select this option, all images are upscaled to a higher resolution to lower the amount of broken dots.



• **Full Resolution**: if you select this option, all images are upscaled to the full ripping resolution. This gives the best results, but comes with the greatest performance cost.



- **Note:** For RIP resolutions below 2000 dpi, **Automatic** and **Full Resolution** are basically the same.
- **Low-Pass Resample**: if you select this option, the resolution of the image does not change. In stead, a filter is used on the image that causes a blur. Consequently, the color values are closer to each other and the transition of the dots is less abrupt. This comes with a image quality cost.



# **Separation Specific Settings**

- Override defaults: here you can give extra options in the separation list underneath.
  - Set ruling per separation: if you select this checkbox, you can define a specific ruling for each separation in the separation list underneath.
  - Set dotshape per separation: if you select this checkbox, you can define a specific dotshape for each separation in the separation list underneath.
- Empty separations
  - **Output empty process colors:** if you select this checkbox, a TIFF file will also be generated for process colors inside a file that don't contain information (for example in a duotone file).
  - Spot color screens: here you can define what should be done with the spot color screens. Options:
  - **Don't output spot colors**: if you select this option, the spot colors will not be output.
  - Iterate over process color screens: if you select this option, the same angles of the process colors will be used for spot colors.
  - Specify explicitly if you select this option, you can set the angles yourself in the separation list underneath.

- Angle family: here you can select the angle family. Options:
  - Choose to populate: if you select this option, you can define the angles yourself in the separation list.
  - Offset: if you select this option, the predefined offset angles will be entered in the separation list.
  - Flexo: if you select this option, the predefined flexo angles will be entered in the separation list.

**Note:** You can choose which separation you want to process or not by selecting/deselecting the checkboxes in the **Print** column in the separation list.

# Calibration

#### **Dot Gain Curves**

- Enable Curve 1: here you can select the reference curve.
- Enable Curve 2: here you can select the compensation curve.
- Enable Curve 3: here you can select the bump curve.
- **Important:** The order of the curves is important for processing. The order is shown at the bottom of the window: Application Order: input % > 1 > 2 > 3 >Output %
- **Note:** In some cases the calibration is split up in a plate curve and a press curve. In that case: curve 2 is the press curve and curve 3 is the plate curve including the bump curve.

#### **Calibration Results**

Select the disclosure button to reveal this option.

**Separation**: here you can select a separation from the drop-down list to display its calibration results from the curve settings that are selected in the RIP.

#### CIP3

- Generate CIP3: if you select this checkbox, the RIP will generate a CIP3 file to send to the press.
- Preview Resolution: here you can define a preview resolution for the CIP3 file.
- Paper Width: here you can define the width (in mm) of the paper that will be used to print on.
  - Inverted polarity: if you enable this option, the file will be output with inverted polarity.
  - Mirrored: if you enable this option, the file will be mirrored.
- Rotation: here you can define a rotation angle.
- **Image Encoding**: here you can define the image encoding.
- Image Compression: here you can define the type of compression.
- Device: here you can define a device for the CIP3 file.

# Curves

In the CURVES tab of the standalone RIP you can:

- Create new curve Collections and curve Profiles (A) containing curve Recipes (B).
- Manage existing curve Profiles and curve Recipes.
- Define the curve **Details** (C).



# **Curve Collections**

With curve Collections you can manage and group simple curves per Media, Device, Customer or per any other item.

#### Manage existing curve Collections

If you select a curve Collection, you can:

- Rename a curve Collection ().
- Delete an existing curve Collection ().

#### **Create a curve Collection**

- 1. Select + in the column Curve Profile.
- **2.** Enter a name for the curve.
- 3. Select the radio button next to Curve Collection.
- 4. Select OK.
- 5. Add a Curve:
  - Select + to create a new curve for this Collection.
  - Select to import an existing curve.

To be used these curves have to be referenced in a Curve Recipe.

# Create a new curve for the curve Collection

- 1. Select +.
- 2. Enter a name.
- 3. Define the Curve Details on page 361 for the curve.

# Import an existing curve for the curve Collection

1. Select

- 2. Select Import CGATS file to import a CGATS file, an X-Rite densitometer format, or select Import PACKZ curve to import a PACKZ curve.
- **3.** Browse to the file to import.
- 4. Select Upload.

# **Curve Profiles**

With curve Profiles you can create and manage curve Recipes based on Resolution, Dot shape, Ruling and Separation or Angle.

# Manage curve Profiles

If you select a curve Profile, you can:

- **Rename** a curve Profile (
- **Delete** an existing curve Profile (**<b>a**).

Select for more options:

- **Duplicate** a curve Profile (<sup>1</sup>).
- Rename a curve Profile (
- **Delete** a curve Profile (**b**).
- Download a curve Profile (<sup>±</sup>).
- Upload a curve Profile (**1**).
- Import a CGATS file, an X-Rite densitometer format.
- **Refresh** the curve Profiles list (**2**).

#### **Create a curve Profile**

- 1. Select +.
- **2.** Enter a name for the curve.
- 3. Select the radio button next to Curve Profile.
- 4. Select OK.
- 5. Add a Curve Recipe.

# **Curve Recipes**

With curve Recipes you can specify which curve should be applied in specific circumstances.

# What is a Curve Recipe

A curve Recipe consists of **ingredients**, which are the parameters you can use building a curve Recipe:

- Resolution
- Dotshape
- Ruling
- Separation
- Angle

One curve Recipe consists of a combination of one or more of these ingredients.

**Note:** The parameters **Separation** and **Angle** are exclusive, as combining a separation name and an angle in the same rule can lead to ambiguity.

#### Manage curve Recipes

Select for more options:

- **Duplicate** a curve Recipe (<sup>1</sup>).
- Rename a curve Recipe (

• Delete a curve Recipe (1).

#### Create a curve Recipe

To create a curve Recipe, follow these steps:

- 1. Go to RIP > CURVES and select a curve Profile for which you want to create a curve Recipe.
- **2.** Select **+**.
- 3. Select and specify the ingredients you want to use to build the curve Recipe.
- 4. Select OK.
- 5. Define the Curve Details for the curve that is assigned to the Recipe.

#### Treebased rule view

By adding curve Recipes, you create a treebased rule view in the user interface:



There is always a **Default** Recipe defined. You cannot delete or rename this Recipe.

Each combination of predefined ingredients makes sure the RIP uses a specific curve during the RIP process.

If you create a new curve Recipe, the system automatically selects the biggest subset to position the Recipe.

If one ingredient is defined in the selected recipe, the new Recipe is added to the selected Recipe.

#### Example

In the above image example, you have created a separate curve for the following situation: 2540 dpi - HXM - 133 lpi. Consequently, you have to create a Recipe with ingredients for 2540 dpi - HXM - 133 lpi. If you do so, the Recipe will be added to the 2540 dpi - HXM branch:



# Rule based system

Curve Recipes follow a rule based system that determines which curve is used during RIP phase. The RIP looks for the Recipe that best matches with the attributes of the file, starting from the top. When a match is found on top level branch, the sub branches are examined to find an exact match. If no exact match is found, the best matching Recipe on parent level will be used. If the Recipes on parent level do not match either, the Default Recipe is used.

# Example



In this example, there are two top level branches:

- One for a resolution of 5080 dpi.
- One for a resolution of 2540 dpi.

Each resolution branch has children with sub branches that are defined by the following ingredients: dotshapes with screen rulings, and in one case also separations. All these ingredient combinations have an assigned curve. This ruleset with an assigned curve is the Recipe. In the example, 20 Recipes are defined, including the Default.

During the RIP phase, the RIP looks for the Recipe that best matches the RIP settings. The curve that is assigned to the Recipe is used to RIP the file. When looking for the best matching Recipe, the system **starts at the top of the tree and works its way down**, until the best match is found. If no match is found, the Default Recipe is used.

RIP settings <sup>8</sup>	Recipe	Info
<ul> <li>2540 dpi</li> <li>Round</li> <li>150 lpi</li> </ul>	A	Recipe A matches all the file attributes.

<sup>&</sup>lt;sup>8</sup> There is a small tolerance on the resolution, ruling and angle to avoid mismatches due to rounding issues.
RIP settings <sup>8</sup>	Recipe	Info
<ul> <li>2540 dpi</li> <li>HXM</li> <li>200 lpi</li> </ul>	В	There is no Recipe that matches all the file attributes of 2540 dpi - HXM - 200 lpi. The best match, starting from the top, is 2540 dpi - HXM.
<ul> <li>2540 dpi</li> <li>Round</li> <li>105 lpi</li> <li>Orange</li> </ul>	С	There is no Recipe that matches all the file attributes of 2540 dpi - Round - 105 lpi - Orange. The best match, starting from the top, is 2540 dpi - Round - 105 lpi.
<ul> <li>5080 dpi</li> <li>HXM</li> <li>120 lpi</li> </ul>	D	There is no Recipe that matches all the file attributes of 5080 dpi - HXM - 120 lpi. The best match, starting from the top, is 5080 dpi - HXM.
<ul> <li>4800 dpi</li> <li>HXM</li> <li>175 lpi</li> </ul>	E	There is no Recipe that matches all the file attributes of 4800 dpi - HXM - 175 lpi. The best match, starting from the top, is the Default Recipe.

The RIP looks for the Recipe that best matches with the attributes of the file, starting from the top and working its way down. Therefore, the order of the Recipes is very important. To reorder the Recipes, or to change the position of a Recipe, drag and drop the Recipe to the desired position. On the left side, a small black arrow shows if the Recipe will be added before, after or in another Recipe after dropping the Recipe. If the drop is accepted, a green sign is displayed. If the drop is rejected, a red cross is displayed.

**Note:** When a section of the incoming file has Object Based screening attached to it, the screening parameters applied to this object determines the curve that will be applied for this object. This is only the case if the object based has no curve applied to it, or if the curve is an identity curve. In all other cases, the curve applied to the object will be the result of evaluating the object based screening parameters.

# **Curve Details**

=

In the curve Details section, you can define the curve details of the selected curve.

# Use a curve as a reference

You can use an existing curve Collection or curve Profile as a reference. This is useful if you need to reuse the same Recipe in multiple occasions. In the curve Profiles list, an icon indicates which profiles are referencing the current Recipe. When a curve Profile is referencing a curve Recipe that doesn't exist, an error icon is displayed in the curve Profiles list.

# Curve Type

You can select if the curve is a **Direct** or **Compensation** curve.

- A direct curve is applied directly to the file.
- A compensation curve is applied to compensate the dot gain on the press. Create the curve by entering the X and Y values in the right table, where X is the desired dot percentage and Y is the percentage measured on the print surface.

# **Curve creation**

• In case of a **direct curve**, create the curve by entering the X and Y values in the right table, where X is the desired dot percentage and Y is the percentage you need to obtain it.

<sup>&</sup>lt;sup>8</sup> There is a small tolerance on the resolution, ruling and angle to avoid mismatches due to rounding issues.

- In case of a **compensation curve**, create the curve by entering the X and Y values in the right table, where X is the desired dot percentage and Y is the percentage measured on the print surface.
- There are two ways to add a curve point:
  - 1. You can add a point in the value table. The curve graph will display a graphical representation of the curve.
    - a. Select + or > Add New Point.
    - **b.** Enter the X value.
    - c. Go to the Y field by pressing the Tab key or by selecting it.
    - d. Enter the Y value.
      - 1. If you press the **Tab** key, the point is added and a new X field is automatically activated where you can start adding a new point.
      - 2. If you press the Enter key, the point is added and you can no longer press the Tab key to add a new point. To add a new point, you have to repeat steps 1 to 4.
      - **3.** Press the **Escape** button or select **x** to cancel.
  - 2. You can click on a position on the curve where you want to add the point. You cannot click on an existing point.
- There are two ways to edit a point:
  - 1. You can select the X or Y field and enter the updated value. Press the **Tab** or **Enter** button to validate the updated value. Press the **Escape** button to cancel.
  - 2. You can drag and drop an existing point on the curve.

**Note:** The 0 and 100 X values are mandatory and cannot be edited or removed.

You can also upload a CSV file that contains curve points. These points will be added to the curve. To do this, drag and drop the CSV file in the user interface or select > Upload Measure.

The CSV file needs to be properly encoded CSV. This implies the following restrictions:

- The separator must be ;
- The values must be between **0-100**
- The file must contain an even amount of numbers, where the odd ones represent the x-values and the even ones the corresponding y-values.
- The following characters are not allowed: "',
- It may not contain different points with the same x-value like ;2;3; and ;2;5;
- **Note:** The curve point have to be strictly ascending. If not, the error message **CURVE NOT VALID** is displayed.

# Highlights

- Zero Output Below: here you can define a percentage below which all output values will be set to zero. This value needs to be smaller than the Minimum Dot.
- **Minimum dot**: here you can define the minimum dot percentage that you want to bump your curve to. The minimum dot is the smallest dot that can be printed before it disappears. It can only be specified if there is no reference curve. In case a reference curve is necessary in the curve preset, a combination of curve presets should be created.
- Smooth Zone Until: here you can define a value until which the curve will be smoothened. When entering zero, no smoothening is applied to the curve.
- Keep 0% at 0%: if you enable this option, the curve will be taken down to 0% at 0%.
- Use PDF input values: if you enable this option, the PDF input values will be used.

# Shadows

• Set Output to 100% Above: here you can define a percentage above which all values will be set to 100%. This value needs to be higher than the Maximum Dot.

- Maximum Dot: here you can define the maximum dot percentage. The maximum dot is the biggest dot that can be printed before it becomes 100%. It can only be specified if no reference curve is set. In case a reference curve is necessary in the curve preset, a combination of curve presets should be created.
- Smooth Zone From: here you can define a value from which the curve will be smoothened. If you enter a value smaller than the Maximum Dot, the curve will be smoothened up till the value set. When entering hundred, no smoothening is applied.
- Keep 100% at 100%: if you select this checkbox, the curve will be pushed up to 100% at 100%.
- Use PDF input values: if you select this checkbox, the PDF input values will be used.

# Description

In the **Description** field you can define a description for the curve.

# **Color Profiles**

In the COLOR PROFILES tab of the standalone RIP you can manage color profiles.

You can:

- Upload a color profiles (\*.icc or \*.icm).
- Remove uploaded color profiles.

## Upload a color profile

To upload color profiles, follow these steps:

- 1. Select Browse files....
- **2.** Browse to the color profile and select it.
- 3. Select Upload. The color profile appears in the list.
- 4. You can now apply it to a RIP preset via the **PRESETS** > Color Management.

Select Clear List to clear the list of uploaded color profiles.

# Logs

In the **LOGS** tab of the standalone RIP you can have an overview of all the RIP messages, categorized by Date, User, Message and Asset. This will give you a detailed overview of the activities in the Standalone RIP.

# **TIFF** viewer

The CLOUDFLOW RIP has an integrated **TIFF viewer** in PROOFSCOPE which allows you to do a comprehensive quality check of the ripped file. To view a file in the PROOFSCOPE TIFF viewer, select it in **JOBS** > **FILES** after the RIP job has been finished.

See View a file with PROOFSCOPE for detailed information on the TIFF viewer.

# SHARE

**CLOUDFLOW SHARE** is a technology to share and synchronize information between separate and independent CLOUDFLOW servers across the globe.

The servers are not part of a cluster. Instead, they are all distinct and have their own databases.

The sharing works over the internet or network. The servers are registered to each other and use **public-private key** encryption to encrypt the communication between the sites.

Use cases of CLOUDFLOW SHARE:

- Share files and information between CLOUDFLOW servers of a large company group located worldwide.
  - Share job folders
  - · Allow multiple operators to work on the same job
  - Archive files to one location
- Share files and information between CLOUDFLOW servers of different companies to deliver files to a supplier or customer (supply chain).

For a good performance of SHARE, you need to take into account several requirements. See SHARE requirements on page 15 for a detailed overview.

# Set up a SHARE network

To share information between different servers, you need to set up a SHARE network between these servers and define which files or folders need to be synchronized.

To do this, follow these steps:

- **1.** Define a server as a site.
- 2. Add the sites to the SHARE network to make them aware of each other.
- 3. Define which files or folders need to be synchronized.



Note: If there are multiple sites connected, the site name will be shown as prefix of the tab's title.

# Define a server as a site

A site is a CLOUDFLOW server that is identified as a site.

You can configure a server as a site in the CLOUDFLOW SHARE settings:

#### THIS SITE

- **Name**: here you can define the name of the site. This information is used when setting up CLOUDFLOW SHARE to identify the site more easily.
- **Description**: here you can define the description of the site. This identification is used when setting up **CLOUDFLOW SHARE** to identify the site more easily.
- URL: here you can specify the URL of the site. This information will indicate how the server can be reached from different nodes in the CLOUDFLOW SHARE setup.

You need to set up the site configuration on all CLOUDFLOW systems that are involved in the CLOUDFLOW SHARE.

# Add a site

You need to add the sites to make them part of the SHARE network.

After having configured each server as a site, you need to make them aware of each other. To make a site aware of another one, you need to add it. This way, it will become a part of the SHARE network.

To do this, follow these steps:

- 1. Open SHARE > SITES. You will see the site that is already configured on this server.
- 2. Select Add Site.
- 3. Specify the Site URL, Admin user and Password of the site you want to add.
- 4. Select Add.

You need to add every site once to another site to make them part of the SHARE network.

=

Note: Once you have added a site to another, you cannot modify its name anymore.



**Note:** If you use HTTPS, make sure that the URLs that are mentioned in the certificates match with the URLs that are used for the sites.

# Create a sync specification

A sync specification is a list of mappings, defining which files or folders need to be synchronized by the SHARE network.

To create a sync specification and add a folder mapping to it, follow these steps:

# 1. Open SHARE > SYNC SPECIFICATION.

- 2. Select Add syncspec.
- 3. Specify a name for the sync specification and select the target site.
- 4. Select Add.
- 5. Hover over the sync specification and click to open the details
- 6. Select Add mapping.
- 7. Complete the following fields:
  - a. File Mapping: here you can define a description for the files you want to map.
  - **b.** Source Site: here you can define the source site. You can set the direction of a Share as either source -> destination (uni-directional) or source <-> destination (bi-directional).
  - **c.** Source Folder: here you can enter the CLOUDFLOW URL of the source folder (for example cloudflow://PP FILE STORE/site a/).
  - d. Target Folder: here you can enter the CLOUDFLOW URL of the target folder (for example cloudflow:// PP\_FILE\_STORE/site\_b/)
  - e. Filter: here you can define a filter if needed. The filter pattern is a Regular expressions on page 475.

**Note:** You can use https://regex101.com as an online regex tester.

8. Select Add.

9. Hover over the syncspec to open it. Possibly you need to refresh the page before you can view the details.

10. Select the Sync Mode. Options:

- **Bidirectional**: if you select this option, changes from both source and target folders are synchronized.
- Source → Target: if you select this option, only changes from the source site are synchronized. Changes at the target site are discarded.
- Target → Source: if you select this option, only changes from the target site are synchronized. Changes at the source site are discarded.
- **Disabled**: if you select this option, nothing is synchronized.
- **Propagate file deletions**: if you select this checkbox, file deletions will also be propagated to the other site. This checkbox is only available if you selected **Source** → **Target** or **Target** → **Source**.

As a result, the mapped folders will now be kept in sync and any file modification is now synchronized. This means:

- Each file modification in the file location in site a, will be synced to site b, and vice versa.
- If a file is deleted in site\_a, the file is also deleted in site\_b, and vice versa.
- If a new file appears in site\_a, this file will also appear in site\_b, and vice versa.

## Remarks:

- In case of a collision (for example if a file is edited simultaneously in both site\_a and site\_b), the source location always wins.
- Deleted files are moved into a **.Trash** folder in the root of the File store to prevent accidental data loss. If desired, files in the .Trash folder that are older than a predefined number of days will be removed. See MAINTENANCE on page 48 for more information.
- With **Remove** you can delete a mapping or a sync specification. As a consequence, all files on the target site will be removed.

# View the share status

When information is shared between sites, you can see the synchronization progress between sites in the STATUS tab in the SHARE tab.

The information is kept available one hour after the synchronization is complete.

# **SHARE** best practices

This chapter provides two use case examples of SHARE best practice setups.

#### Use case 1



Setup

- A CENTRAL CLOUDFLOW in the cloud.
- Two production sites: **SITE 1** and **SITE 2**. Both sites have two File stores:
  - **RW** (Read and Write): this is a File store with read and write permissions and where the operators work on the files.
  - **RO** (Read Only): this is a File store with read permissions only and where the operators save the files when they are finished. This File store is synchronized with the central CLOUDFLOW site through a synchronization mapping.

#### Example of a workflow

All production files are stored on the CENTRAL CLOUDFLOW. Because of the synchronization mapping, all the files are identical on the CENTRAL CLOUDFLOW, File store RO in SITE 1 and File store RO in SITE 2.

- 1. An operator in SITE 1 is working on file A on File store RW in SITE 1.
- 2. When finished, the operator checks in (copies) file A on File store RO in SITE 1.
- **3.** Through the synchronization mapping, file A is automatically updated on the CENTRAL CLOUDFLOW and File store RO in SITE 2 as well.
- 4. An operator in SITE 2 wants to work on file A. To do that, the operator **checks out** (copies) file A to File store RW in SITE 2.
- 5. When finished, the operator checks in (copies) file A on File store RO in SITE 2.
- **6.** Through the synchronization mapping, file A is automatically updated on the CENTRAL CLOUDFLOW and File store RO in SITE 1 as well.

#### Use case 2

In this example there are 2 sites: Site A and Site B. An operator in Site A enters a New Order in the Management Information System (MIS). Consequently, a job folder is automatically created on the File store in Site A, containing the files.

The operator wants to outsource certain jobs to Site B, for example the **Artwork**, **Images**, **pdfQCFiles** and **Technical** jobs. Therefore, the operator creates a Production Order for these jobs in the MIS and marks them to be outsourced. Through the sync spec, the outsourced jobs are synchronized to Site B. This means that the job folders and files of these outsourced jobs are copied to and synced with the File store in Site B. The jobs that are not outsourced are not accessible in Site B.



The sync spec keeps the outsourced jobs synchronized in Site A and Site B. This means that the files of the outsourced jobs are identical in Site A and Site B during the editing phase of the files.



When the operator in Site B is finished, the job is marked as **done** in the MIS in Site B. Consequently, the folders that were created on the File store in Site B are deleted as soon as the synchronization of the files is finished. The updated files are now only accessible in Site A.



# DATALINK

With this module you can connect to a database.

It consists of two parts:

- A tab in CLOUDFLOW that you can use in a standalone environment.
- The data nodes.

# **Connect a database**

In the DATALINK tab you can connect your database with CLOUDFLOW.

To connect a database, follow these steps:

# 1. Activate the JAVA WEB APPS HOST Worker.

This Worker hosts java web archives. Go to **SETTINGS** > **WORK SERVERS** to activate it. See JAVA WEB APPS HOST on page 44 for more information on the Worker and on how to deploy war files.

# 2. Add your database

- 1. Go to DATALINK > DBs.
- 2. Select ADD A DATABASE.
- **3.** Complete the following fields:
  - Name: here you can define a name for the connection.
  - **Driver**: here you can define the driver. The driver depends on the type of database you want to connect. You can copy the correct driver from the table below.
  - **JDBC URL**: here you can enter the JDBC URL. This also depends on the type of database and can be copied from the table below. Make sure to replace **host** and **dbname** by the correct values.

## Example

You have a postgress database called **mydba** and your host is **127.0.0.1**. In that case, the JDBC URL would be jdbc:postgresql://127.0.0.1:5432/mydba. If you're using a local host, jdbc:postgresql://localhost:5432/mydba is also possible.

=

**Note:** Postgres uses 5432 as the default port. If you change the port, you also need to change it in the JDBC URL.

- Username: here you can enter the username.
- **Password**: here you can enter the password.
- Max Total: here you define the maximum number of database connections that can be opened at any time.
- Max Idle: here you can define the maximum number of idle database connections.
- Min Idle: here you can define the minimum number of idle database connections.
- Test Query: here you can enter a test query.
- Default Transaction Isolation: here you can select the default transaction isolation.
- 4. Select SAVE.

All connected databases appear in an overview list.

- To edit an existing connection, select the connection.
- To test the connection, select
- To delete the connection, select

# Add a query

In the DATALINK tab you can add database queries.

To add a database query, follow these steps:

- 1. Go to **DATALINK** > **Queries**.
- 2. Select ADD A QUERY.

- **3.** Fill in the following fields:
  - Name: here you can define a name for your query.
  - DB Connection: here you can select the database connection for which you want to create a query.
  - Query: here you can enter your query.
  - **Default clause**: here you can define the default clause.
  - Default Order By: here you can define the default order by.
  - **ID column name**: here you can define the ID column name.
  - **ID parameter name**: here you can define the ID parameter name.
  - Cast Expression: here you can define the cast expression.
  - Trim Result: if you select this checkbox, the spaces at the end of the result strings will be trimmed.
  - Resulting Column Name Patch
    - Regular Expression: here you can define a regular expression on the resulting column names.
      - **Note:** You can use https://regex101.com as an online regex tester.
    - Replace value: here you can define a replace value for the regular expression.
  - Save Query: here you can define a save query.
- 4. Select SAVE.

# JOBS

In CLOUDFLOW JOBS you can manage Jobs.

#### What is JOBS

JOBS is a CLOUDFLOW module that can handle Jobs. You can create new Jobs, manage existing Jobs and interact with them.

JOBS is integrated in the workflow engine. It is therefore very flexible and it can be linked with your production flow.

JOBS comes with a predefined user interface that cannot be changed.

# What is a Job

A Job is a collection of different information types that belong together. It contains data and assets.

#### Data:

- Job Name
- Status
- Description
- Custom data (such as customer name, printing process...)
- ..

All Jobs data are stored in the database in a dedicated Jobs collection.

#### Assets:

- Files
- Folders

# How does it work?

There are dedicated nodes for the actions you want to perform with JOBS. See Jobs on page 247 for an overview of all the Job nodes.

Working with JOBS comes down to three main concepts:

#### • Creating a Job.

A Job is created by the Create Job on page 248 node.

Anyone (with the right permissions) can create a Job. It is not limited to internal employees of your company. For example, you can easily set up a web portal for your customers or suppliers and give them the possibility to start a workflow that creates a Job.

• Interacting with an existing Job.

Interactions with an existing Jobs are also driven by specific nodes in a workflow. For example, the Update Job on page 251 node allows you to update the status of a Job or add additional files.

• Managing an existing Job.

There are various possibilities to overview and manage existing Jobs:

- Since Jobs are driven by a workflow, you can use KIOSK.
- You can use the Job list.
- You can use **Form Builder** or **Pagebuilder** to build a web portal. This way, external parties can both start and manage their Jobs.

## What can you use it for?

JOBS is a very versatile module in CLOUDFLOW and can be used for many application fields. For example:

- In a prepress environment where prepress operators use JOBS to add prepress information such as metadata, file information... to a Job. The operator can use **KIOSK** to browse and manage the Jobs.
- In a more extended environment where different parties such as operators, customers, suppliers... are involved. Customers could use an external tracking portal to place an order, while internal parties could use it as an internal tracking portal.



**Important:** The default JOBS user interface cannot be changed. If you want to extend the use of JOBS with custom portals, this is outside the scope of the default JOBS user interface. See Use case for a an example of a such a JOBS use case.

• ...

=

Note: The API contains functions dedicated for Jobs (job.\*). They are available on http:// <server>:<port>/?api#job.

# Job workflow

A Job workflow automatically creates a Job.

A Job workflow is built in CLOUDFLOW WORKSPACE and contains a Create Job node. The Job workflow defines the Job properties.

## Create a Job workflow

A Job workflow automatically creates a Job when it is triggered. To create a Job workflow, follow these steps:

- 1. Open the FLOWS tab in CLOUDFLOW.
- 2. Select WORKFLOWS.
- **3.** Select + to add a workflow.
- 4. Enter a name for the workflow.
- 5. Build the workflow by dragging nodes to the workflow area. Make sure to:
  - Use Start From Kiosk as the start node and enter Jobs in the Category field.
  - Add the Create Job node to your workflow. For more nodes dedicated to Jobs, see Jobs nodes.

# Start a Job workflow

There are two ways to start a Job workflow:

- 1. Via the Job list.
- 2. By submitting a job to it in KIOSK. See Submit a job to a workflow on page 258 for more information.

# Job data

Jobs contain data, such as name, files, Job ID, custom data...

These data can be standard Job data, or custom Job data.

#### Standard Job data

Standard Job data are saved to the Job via the **Create Job** node. They are added to the database at Job's data root level.

#### Example

Create Job	? >
Template Job ID:	
Parent Job ID:	
Job Type:	
Identifier:	
Name, Job some	
State: Job_name	
Description:	
•	
Files:	
Files Tag:	
	Cancel Save

This will result in the following JSON object in the database:

"state"		:	"Joł	С	stat	te'	۰,
"name"	:	'	'Job	n	ame'	",	
}			-	_			

# **Custom Job data**

Custom Job data should always be saved to the Job in the custom section. You can do this in **Create Job** > **Define** (custom) data to be assigned to the created job. You need to define a JSON path that selects an element in the JSON document and a value that you want to set for the specified data path.

#### **Example 1**

Path	Value
custom.cusname	CustomerX
custom.cusID	12345
identifier	Job_442542

This will result in the following JSON file in the database:

```
{
    "identifier" : "Job_442542",
    "custom" : {
        "cusname" : "CustomerX",
        "cusID" : "12345",
    }
}
```

Note: In case the fields cusname or cusID do not exist, they will be created.

You can also leave the Path empty. In this case, the data are sent to the root level of the Job data in the database.

# Example 2

╤

Suppose you have created the following Job form in Handle Form on page 250:

Custome	r Info	rmation	
		JobID	78910
		Customer Name	ClientA
		Customer ID	3636
	FILES		
	Upload	Cloudflow	
	Browse file	16.	
	browse file	79	
			Submit

Where:

- The JobID key is identifier.
- The Customer Name key is **custom.cusname**.
- The Customer ID key is **custom.cusID**.

In Start From Kiosk > Job Data Variable Name, you have specified a variable for the Job data, called project\_data. The variable will have this content:

```
{
  f
  f
  project_data: {
    source: "jacket",
    value: {
        identifier: "78910",
        custom.cusname: "ClientA",
        custom.cusID: "3636"
  }
}
```

Where:

- identifier is a direct parameter.
- **custom.cusname** and **custom.cusID** are custom parameters. It is important to use the dot notation **custom.X** to add the custom data in the custom section.

You can leave Path empty and define the variable in Value:

Path	Value
	project_data

In this case, the data in of the Job form (that are in the variable) are sent to the root level of the Job data in the database. This results in the following JSON in the database:

```
{
    "identifier" : "78910",
    "custom" : {
        "cusname" : "ClientA",
        "cusID" : "3636"
    },
}
```

=

**Note:** In this example, the **identifier** value can also be set in the **Create Job** node itself (in the **Identifier** field). In this case, and in all other cases where there are values in both the direct parameters (coming from the node fields) and the Job form data, the values of the direct parameters will always be used and the data in the variable will be overwritten.

#### Custom properties per file: best practice

In case you want to add custom properties for each file in the Job, it is best practice to use a nested job object for each file. In that case, one parent Job contains a child Job per file, and you can save the custom data inside the child Jobs:

Parent Job

- Child Job 1
  - File data
  - Custom data
- Child Job 2
  - File data
  - Custom data

# **User Interface**

The JOBS module comes with a predefined user interface that cannot be changed.

In the JOBS UI you can do the following:

- Create and manage Job configurations.
- Create a Job.
- Manage existing Jobs (for example, add files, change the status...).
- Have an overview of the tasks in a Job.

# CONFIGURATION

In the Jobs configuration you can create and manage Job configurations.

A Job configuration contains the Job settings per Job type. For each Job type configuration, a label appears in the JOBS topbar. For example:



To open the Job type configurations, select JOBS > CONFIGURATION. The tab consists of two sections:

- General
- Job types

## General

In this section you can define the general settings for the Job lists in the Job Dashboard.

- # rows: the number of rows in the Job list.
- Kiosk: if you activate this option, the taskbar shows an access to KIOSK.

## Job types

In this section you can manage and add Job type configurations.

Select **t** to add a Job type.

Select the disclosure icon you to show or hide the parameters of a Job type.

## General

- Name: the name of the label that is displayed in the JOBS topbar. If you select this label in the JOBS topbar, the Job Dashboard of a specific Job type opens.
- Job type: the Job type. This value needs to correspond with the Job type in the Job workflow.
- Order by: the default order of the Jobs in the list.
- Root tag: the specified tag of the assets root folder. This tag makes sure that the assets folder is visible in the Assets section in the Dashboard. The tag you specify here is the tag that you need to enter in the Folder Tag in the Create Job node in your Job workflow.
- Row Height: the row height of the rows in the Jobs list in the Dashboard.
- **Image Size**: the image size.
- Views?: if you activate this button, the Views are activated in the Job Dashboard. This means that an arrow appears next to the label in the JOBS topbar that allows you to select a created View in the drop-down list. The Job list is filtered and only the Jobs that match with the View parameters are listed. See Searches and Views for more information about Views.

- Search properties: the search properties allow you to define database keys and labels. The keys are used to search for the Job data in the database. Select 🔂 to add a search property:
  - **Property**: the key that is used to search for the Job data in the database. See Job data on page 374 for more information about Job data.

### Examples

- **state** is a Job data entry at root level. This search key will search for Jobs with a particular state in the Job data.
- You can also use custom Job data, such as **custom.cusname**. This search key will search for Jobs that belong to a customer with a specific name.
- Label: the label of the search that will be displayed in the UI.
- **Type**: the type of search.

## Whitepapers

Here you can define the use of the workflows in the Job type.

- Create?: if you activate this button, you will be able to create a Job in the Job Dashboard.
  - Whitepaper: the workflow that will be used to create the Job.
  - Input: the input name of the Job workflow that will be used to create the Job.
- File handler: the workflow that handles the files.
  - Whitepaper: the workflow that handles the files. This workflow is needed to handle (for example delete, add...) files via the Assets section in the Dashboard.
  - Input: the input of the workflow that handles the files.

# Dashboard

The Job Dashboard gives you a list of the Jobs of a specific Job type and allows you to create a Job of a specific Job type.



• A Topbar: to open the Dashboard of a specific Job type, select the corresponding label in the topbar. See CONFIGURATION on page 377 for more information on how to create a configuration for a specific Job type.

• B

- View column buttons: select one of the buttons to view the selected Job, the Item or the Job of item details using one , two of three columns. If you view
- Breadcrumb: the breadcrumb displays the navigation trail of the selected Job or Item.

#### Example



In this example, the details of Item ID **643434343**, which is an Item of Job ID **774435** are displayed. The view is split into three windows, which means that the following is displayed:

- Column 1 displays the Job data of the selected Job (774435) in the Job list.
- Column 2 displays the Job details of the selected Job (774435).
- Column 3 displays the Item details of the selected Item (774435).
- C Job data of the selected Job. See Job data on page 374 for more information.
- **D** Job details of the selected Job. See Job details on page 380 for more information.

#### Job list

In the Dashboard you can see the Job list of a specific Job type.

#### Job list

The Jobs are displayed in the Job list. In case you have used a search query, the list of Jobs that match the query is displayed. See Searches and Views on page 381 for more information about search queries.



**Tip:** If only a part of the Jobs are listed, check if there is no search query activated. If there is, your Job list is filtered. To display all Jobs in the Job list, delete the search query and select **Search** again.

#### Job data

If you have specified a form for the Job list, the Job data for each Job are displayed in the Job list.

You can specify this list form in Create Job > Form name for 'list edit'. Steps:

1. Create a form in your Job workflow with the Handle Form node.

# 2. Add the form to the Create Job node in your Job workflow: enter the input name of the form in the field Form name for 'list edit'.

If you want to update the Job data in the list form, follow these steps:

- **a.** Select the field you want to update.
- b. Update the content. Select Revert to reset your edits.
- c. Select Save.

#### Job details

If you select a Job, the Job details are revealed on the right.

Depending on your Job workflows, the Job details includes multiple sections:

774435 -	
> Form	🖱 Revert 🛛 🖉 Save
> Assets	
> Tasks	
> Items	+
<ul><li>&gt; Items</li><li>&gt; Automation</li></ul>	+ Create New Item
<ul> <li>&gt; Items</li> <li>&gt; Automation</li> </ul>	+ Create New Item

MARS contains a MARS App called **JOBS-Labels Starter Kit**. This App contains workflows that you can use to create your initial JOBS workflows for a label printer and that result in a Job with the following sections in the Job details:

## Form

If you have specified a form for the Job details, the Job details of the selected Job are displayed on the right. You can specify this Job details form in **Create Job** > **Form name for 'form edit'**. Steps:

- 1. Create a form in your Job workflow with the Handle Form node.
- 2. Add the form to the Create Job node in your Job workflow: enter the input name of the form in the field Form name for 'form edit'.

#### Assets

This section lists the assets and the assets folders in the Job.

You can use a contextual menu that you can use on a selected folder or file:

- If you have selected a folder:
  - Upload file: to upload a file in the selected folder. You can also drag and drop a file.
  - Rename folder: to rename the selected folder.
  - Open folder: to open the folder.
  - Show in portal: to show the folder in the Asset details.
  - Submit to workflow: to submit the folder to a workflow.

- If you have selected a file:
  - Copy to clipboard: to copy the file to your clipboard.
  - **Rename file**: to rename the selected file.
  - **Remove file**: to remove the selected file.
  - **Reveal file**: to open the folder where the file is located.
  - Show in portal: to show the file in the Asset details.
  - **Open in application**: to open the file in the application defined in the CLOUDFLOW Plug-in Suite SETTINGS.
  - View in proofscope: to view the file in PROOFSCOPE.
  - Submit to workflow: to submit the file to a workflow.
- **Note:** The workflow specified in **JOBS** > **CONFIGURATION** > **Whitepapers** > **File handler** controls how to handle the files that are added.

#### Tasks

This section lists the tasks in the Job. This can be for example a manual prepress step represented by a workable stop in the **Hold In Kiosk** node. You can add tasks to a Job with the **Add Task to Job** node.

Depending on their states, tasks can be displayed in various colors:

- Pending: light blue.
- Started: light green.
- Ready: dark green.
- Done: dark blue.

#### Items

This section you can:

- See an overview of the existing Items in the Job. Select the Item to reveal the Item details on the right.
- Create a new Item. To do so:
  - 1. select Create New Item.
  - 2. Enter the Item parameters.
  - 3. Select Submit.

#### Automation

This section displays the automation flows behind the Job.

#### **Searches and Views**

You can create Views and Searches in the Dashboard to easily filter Jobs.

# Create a View or a Search

To create a view/search, follow these steps:

- 1. Select the Job type for which you want to create a view or a search.
- 2. Select OAdd.
- **3.** Enter the query parameters. The content of the drop-down list depends on the **Search properties** that you have configured for this Job type.
- 4. Select **Save** to save the query.
- 5. Enter a name.
- 6. Select whether you want to save the query as a View or as a Search.
- 7. Select Save.

The View or the Search query is now available in a drop-down list in the Dashboard.

#### **Difference between Search and View**

If you select a View, the query is completed and the Jobs that match the query will be listed immediately.

If you select a Search, the query is not completed. You have to enter the search value and select **Search** before the Jobs that match the query are listed.

#### Delete a View or a Search

To delete a View or a Search, select the View or Search from the drop-down list and select Delete.

#### Create a Job

To create a Job via the Job Dashboard, follow these steps:

- 1. Select +.
- 2. Enter the Job parameters in the Job form.
- 3. If needed, upload the files.
- 4. Select Submit.

The Job is now created and appears in the Job list of the Dashboard.

# Jobs in KIOSK

Operators can view Job data in KIOSK.

d	ĥ	LOUDFLOW		ALL SCOPES . Y	HELLO ADMINISTRATOR
٠	3	<b>П</b> кіовк			
Q		A	X Error III Hold > Running > Finished Modified Last 24-hours 1 Al Workflows	e) 🔺 💟	
	C	Create New		Demo design_icc	
-	1	Demo design_icc     Demo design_icc	Notey 22320 FM (* 😫	III Demo design_icc	
<b>)</b> **		III Demo+design Demo+design	B today 2:02:48 PM	State To be reviewed	
1	-	III Demo+design Demo+design	today 2:02:41 PM	Description Job with reference 434645425435	
<b>)</b> **	Ø,	III Job1 Job1	today 2:01:13 PM	FILES	
1	Ø,	III Job1 test_icc	today 2:01:07 PM	Demo design_loc.pdf	
ø		DC2254-process	6/21/2017, 12:30:04 PM		
×		DC2254.cf2	6/21/2017, 12:29:37 PM		
×		DC2254-process	6/21/2017, 12:27:51 PM		

- A: You can search Jobs by name, status, description and the name of the input files.
- **B**: You can view the Job info, so the operator can see the status of a Job workflow.
- C: You can view the Job details.

# **JOBS** examples

# Example of JOBS workflows for a label printer

An example of how JOBS can be used by a label printer.

MARS contains a MARS App called **JOBS-Labels Starter Kit**. This App contains workflows that you can use to create your initial JOBS workflows for a label printer.

# Use case

JOBS have multiple and versatile application fields.

The following is an example of a use case with the JOBS module.

# **Parties involved**

Who is involved?

# The brand owner

**Chip** works as a project manager at **Reynholm Industry Brands**, a brand owner that among others produces energy drinks. The company works together with various suppliers such as design, marketing and translation offices to have their packaging created and translated, and printing companies to have it printed.

Reynholm Industry Brands' marketing partner has just delivered an updated packaging design for the most popular canned energy drink product, **Star Drink**. Before ordering the offset printing job at their long-term printing partner, Chip wants to order some printed proof first.

# The printing company

**Dale** is head of production at **Wernham Hogg Offset**, an offset printing company. The company is one of the suppliers of Reynholm Industry Brands.

## Actions taken

What do they do?

# Chip

To place the order, Chip takes the following steps:

- 1. He logs in on the customized order portal of Wernham Hogg Offset with his own username and password. He lands on a personal and private page for that was set up by Wernham Hogg Offset for Reynholm Industry Brands where it shows the status of Chip's submitted jobs.
- 2. For a new job, Chip selects the Submit Job button.
- **3.** He fills in the requested parameters:
  - **a.** Job ID: the Job ID. This can be a manual ID or a number coming from an MIS system. In this case, Chip will use a manual ID, **25042017**.
  - b. Customer name: the customer name, in this case Reynholm Industry Brands.
  - c. Job Description: a description of the job for quick reference. In this case, Chip will use a description of New Star Drink design.
  - d. Run Quantity: the amount of proofs Chip wants to order. For this order, chips needs two proof copies.
- 4. He browses to the design file that needs to be processed, uploads it and selects Submit.
- 5. In the order overview, Chip sees the newly submitted order. It has status **IN REVIEW**, which means that Wernham Hogg Offset still has to give feedback whether they can provide the requested job or not.
- 6. If Wernham Hogg Offset accepts the job, Chip sees the job in a status of IN PRODUCTION, INTERNAL QC and eventually, OUT FOR APPROVAL.
- 7. At the **OUT FOR APPROVAL** status, Chip can select the View button to review the proof in PROOFSCOPE prior to Wernham Hogg Offset generating the final proofs and either **Accept** or **Reject** the approval.
- 8. Once approved, Wernham Hogg Offset will produce the proofs.

## Dale

As soon as Chip has placed the order, Dale sees a new job appearing in **KIOSK**. If desired, he can open the file in **PROOFSCOPE**.

Dale has to make a decision:

• If he decides the file is OK for production, he selects the button that says OK, TO PRODUCTION.

• If he decides the file is not OK for production, he selects the button that says **NOT OK, BACK TO CUSTOMER**.

Depending on Dale's decision, the Job in Chip's order overview has changed:

- If Dale has decided to start production, the Job's status is updated to IN PRODUCTION.
- If Dale has decided that production cannot be started due to a certain reason, the Job's status is updated to **ERROR**.

As Dale continues the job, it flows through the CLOUDFLOW workflow where work is performed and stops at an **INTERNAL QC** state. After Dale's QC department performs the quality control, a PROOFSCOPE approval is started and Wernham Hogg Offset waits for Chip to approve it before outputting proofs.

# Behind the scenes

How is it done in CLOUDFLOW?

#### The order portal = customized HTML page

The order portal where Chip places his order is an HTML page that was customized for Reynholm Industry Brands with PAGEBUILDER. It is protected with a scope so that Chip can exclusively land on a private page that is restricted from navigating to other CLOUDFLOW pages.

When Chip opens the page, he fills in the form and submits the file. This initiates a workflow that creates a Job.

See Pagebuilder on page 66 for more information on how to build custom HTML pages with integrated possibilities to start a workflow.

#### The workflow

The Start From Kiosk workflow (Submit Job) in this use case is built up as follows:



The nodes are configured as follows:

## Start From Kiosk

Named **Start From Kiosk (Submit Job)**. This node makes sure that the workable is created when Chip uploads a file and selects **Submit**. It has the following specified parameters:

- Name: Request For Proof.
- Category: Jobs.

This will make sure the system creates a Job.

# KIOSK parameters:

- A required Text field with (variable) Key jobid and Label Job ID.
  - This will create a variable **jobid** when Chip enters the requested parameter **Job ID** on the order portal.
- A required Text field with (variable) Key customerName and Label Customer Name.

This will create a variable **customerName** when Chip enters the requested parameter **Customer Name** on the order portal.

• A required Text field with (variable) Key jobDescription and Label Job Description.

This will create a variable **jobDescription** when Chip enters the requested parameter **Job Description** on the order portal.

• A Number field with (variable) Key runQuantity and Label Run Quantity.

This will create a variable **runQauntity** when Chip enters the requested parameter **Run Quantity** on the order portal.

- Number of files: One.
- Enable upload: selected.

This enables Chip to upload the file that will be processed by Wernham Hogg Offset.

# Create Job

This node makes sure a Job is created when Chip uploads a file and selects **Submit**. It has the following specified parameters:

• Job Type: PageBuilderExercise.

We will use this Job Type later to filter on this type of job.

• Name: customerName-jobid.

The name of the Job is built with two variables:

- **customerName**: a custom variable created when Chip enters the Customer parameter in the order portal. It is a consequence of the **KIOSK** parameters in the **Start From Kiosk** node.
- **jobid**: a custom variable created when Chip enters the Job ID parameter in the order portal. It is a consequence of the **KIOSK** parameters in the **Start From Kiosk** node.
- State: IN REVIEW.

This will set the Job's initial state to IN REVIEW.

• Description: jobDescription.

A custom variable created when Chip enters the Job Description parameter in the order portal. It is a consequence of the **KIOSK** parameters in the **Start From Kiosk** node.

• File: All files from node previous Node.

The variable **All files from node previous Node** will make sure that a small preview icon of the file is shown when viewing the **Job Details**.

• Define (custom) data to be assigned to the created job:

will store the custom data that Chip entered when submitting the Job to the database document for the job:

РАТН	Value
custom.jobId	jobId
custom.customerName	customerName
custom.jobDescription	jobDescription
custom.runQuantity	runQuantity

## Set Variable

Named Set Variable Job ID. This node will set a variable in the workflow for the current Job.

Name	Туре
newJobId	Job ID

#### Hold in KIOSK

This node will make sure that Dale can decide in KIOSK whether the file is OK for production or not. It has the following specified parameters:

• Viewable files: All files from node previous Node

This will make sure that the file can be opened in PROOFSCOPE.

• Routing labels:

# • OK, TO PRODUCTION

This will create a decision button **OK**, **TO PRODUCTION** in KIOSK and make sure that the workable is directed to the **Job in Production** node when Dale selects this decision button.

# • NOT OK, BACK TO CUSTOMER

This will create a decision button **NOT OK, BACK TO CUSTOMER** in KIOSK and make sure that the workable is directed to the **Job in Error** node when Dale selects this decision button.

#### Update Job

Named **Set Job to Production**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

• Activity: Set State

This will make sure the state is changed.

• State: IN PRODUCTION

This will set the state to **IN PRODUCTION** in Chip's order overview. With this information he knows that his order will be sent to production.

## **Update Job**

Named **Set Job to Error**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

Activity: Set State

This will make sure the state is changed.

State: ERROR

This will set the state to **ERROR** in Chip's order overview. With this information he knows that his order will not be sent to production.

# Script:

Named **Set Files To Be Processed Variable**. This node will set a variable **filesBeingProcessed** to the file originally uploaded by Chip.

```
var filesBeingProcessed = getParameters().files;
setResultVariables({filesBeingProcessed:filesBeingProcessed});
```

# Wait

Named **Do Some Work**. This node waits 10 seconds to simulate work being done. It allows Chip to see the status of the Job change from **IN PRODUCTION** to **INTERNAL QC**. It has the following specified parameters:

• Delay: 10 seconds.

# Update Job

Named **Set Job to Internal QC**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

• Activity: Set State.

This will make sure the state is changed.

• State: INTERNAL QC.

## Hold in KIOSK

This node will make sure that Dale's QC team reviews the file before routing for approval. It has the following specified parameters:

• Viewable files: All files from node previous Node.

This will make sure that the file can be opened in PROOFSCOPE.

• Routing labels: Internal Approval.

# Start Approval

After the Internal Approval happens, this node will begin a PROOFSCOPE Approval for Chip. It has the following specified parameters:

• File to approve: All files from node previous Node.

This will make sure that the file can be opened in PROOFSCOPE.

- Variable name: approvalInformation.
- User Names: Chip
  - **Note:** A user named Chip needs to exist in CLOUDFLOW.

## Select Job

This Node selects the Job to update. It has the following specified parameters:

- Select by: Job ID.
- Job ID: newJobId (this is the variable set earlier in Set Variable Job ID node).

## Update Job

This node will set some data for the job, like the Approval ID and Approval user that we will use later in PAGEBUILDER. It has the following specified parameters:

РАТН	
custom.approvalId	approvalInformation.id
custom.approvalUser	Chip
custom.filesBeingProcessed	filesBeingProcessed

# | JOBS | **388**

#### **Update Job**

Named **Set Job to Out For Approval**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

• Activity: Set State.

This will make sure the state is changed.

• State: OUT FOR APPROVAL.

# **Route After Approval**

This node will wait for the approval to be accepted or rejected. It has the following specified parameters:

• Policy: All Participants need to Accept (in this example, Chip is the only participant).

#### Update Job

Named **Set Job to Approved**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

• Activity: Set State.

This will make sure the state is changed.

• State: APPROVED.

#### Update Job

Named **Set Job to Rejected**. This node will make sure that the status of the Job is changed to the correct state. It has the following specified parameters:

· Activity: Set State.

This will make sure the state is changed.

• State: REJECTED.

#### The workflow

The **Submit From Form** workflow **Get Approvals** workflow which Pagebuilder calls to get a list of approvals for Chip's in this use case is built up as follows:



The nodes are configured as follows:

## Start From Web Request

Named **Start From Web Request (Get Approvals)**. This node gets the approvals for the job and returns them to Page Builder to display. It has the following specified parameters:

- Name: GetApprovals.
- Workable name: GetApprovals\_id.

This will name the workable with GetApprovals\_ followed by the id number.

# Script

This node gets the list of approvals for the job and sets a result variable. It has the following specified parameters:

- Input Files: (blank).
- Script:

# Set HTTP Reply

This node returns the result variable back to Pagebuilder so it can display the results to Chip. It has the following specified parameters:

- Status: 200.
- Contents: result.

This is the variable set from the Script node.

• Delay: 10 seconds.

# 3D

# 3D

The 3D option in CLOUDFLOW provides several automated 3D options.

You can:

- Render a view of an ICD3D file and save it.
- Replace/update or render labels in an ICD3 file.

The automated 3D actions are available through workflow nodes. See the 3D nodes for more information.

# API

The CLOUDFLOW API exists in various forms.

- As a JSON-RPC
- As a JS API
- As a PHP API
- As a cURL call

You can find the API reference on http://<server>:<port>/?api

# **JSON-RPC**

The CLOUDFLOW API uses JSON-RPC to communicate with the back end, a remote procedure call protocol which uses HTTP as the transport protocol and JSON as the encoding.

All calls are handled by **portal.cgi**, so in order to use the API in another application, you have to send a JSON object to **portal.cgi**. This will establish a data connection during which you can invoke API methods.

All API calls are handled by **portal.cgi**, which is accessible through http://<server>:<port>/ portal.cgi.

To invoke a remote method, you need to send an API request which is replied to with a response.

#### Request

To invoke a method, you need to send a request to **portal.cgi**. A request is a single JSON object and contains the following properties:

- The method ('method'): the method name.
- Parameters ('param'): the parameters.

It looks like this:

```
{
'method' : 'method name',
'param 1' : 'the first parameter',
'param 2' : 'the second parameter'
...
}
```

#### Response

After the method has been invoked, the service sends a response. A response is also a single JSON object and contains the following properties:

• The result ('result'): the returned object.

In case of an error when invoking the method, the service returns an error object with the following fields:

- 'error code': a category code ('generic' in most cases).
- 'error': the error description (usually for debugging purposes).

The following 'error codes' are typically used:

- 'generic': a generic error.
- 'invalid session': the session code is invalid or the session has expired.
- 'no license': the required license is not activated on the machine.
- 'no permission': the user doesn't have permission to call the method with the supplied parameters.
- 'unknown command': the method identifier is not recognized by the server.

#### Parameters for authentication and session

For authentication reasons the JSON object requires at least the following parameters:

• 'session': the session ID.

To invoke a method, the application first needs to obtain a session ID. You can create a session ID by invokingauth.create session. This call requires the following parameters:

- method:auth.create\_session.
- user\_name: the user name.
- user\_pass: the password.

If the call is successful, the server returns a JSON object with a 'session' field. If it fails, the return value is a error of type 'generic'.

A session is valid for 12 hours.

- 'session\_scope': an optional parameter indicating the user scope. The default value is \*. If the user has only one scope, the parameter is ignored.
- 'method': the name of the method to invoke, in form 'module.method' (for example 'proofscope.create url').

# **JS and PHP layers**

For ease of use, there are two thin layers available on top of the JSON-RPC over HTTP API.

- 1. A JS API, which provides each JSON call as a JavaScript function. It formats the JSON message and performs the AJAX call. It is accessible on each CLOUDFLOW installation at http://<server>:<port>/?api=js.
- 2. A PHP API, accessible on http://<server>:<port>/?api=php.

Both JS and PHP API references are accessible and documented on http://<server>:<port>/?api

# cURL

cURL is a command line tool that can perform HTTP requests (https://curl.haxx.se/). With this tool the low level JSON over HTTP API of CLOUDFLOW is exposed.

To perform an API call using cURL, follow these steps:

#### 1. Retrieve the contents of an HTTP request with a JavaScript command.

In order to perform an API call with a cURL command line, you first need to inspect the content of the HTTP request with a JavaScript command. The returned data are then used in cURL.

To do this, follow these steps:

1. Log in to CLOUDFLOW with Google Chrome.

The reason for this is that the CLOUDFLOW UI uses the API to show the pages, so you can perform API calls with the JavaScript console. Google Chrome is a perfect browser for this.

## 2. Open Developer Tools > Network.

This will give you an overview of the network calls in the JavaScript console:

2000 ms 4000 ms 600	10 ms	8000 ms	10000 ms	12000 ms	14000	0 ms 16000 ms	18000 r
Name	Status	Туре	Initiator	Size	Time	Timeline – Start Time	
portal.cgi	200	xhr	jquery-1.8.3	7.3 KB	74 ms	1	
portal.cgi	200	xhr	jquery-1.8.3	840 B	71 ms	1	
portal.cgi	200	xhr	jquery-1.8.3	1.5 KB	51 ms	1.1	
_ portal.cgi	200	xhr	jquery-1.8.3	202 B	74 ms	1	
iii filestore.png	200	png	Other	895 B	2 ms	- I -	
portal.cgi	200	xhr	jquery-1.8.3	202 B	46 ms	- E	
portal.cgi	200	xhr	iquery-1.8.3	7.3 KB	50 ms		
portal.cgi	200	xhr	iquery-1.8.3	840 B	47 ms		
portal.cgi	200	xhr	iquery-1.8.3	1.5 KB	70 ms		
portal.cgi	200	xhr	jquery-1.8.3	202 B	70 ms		
portal.cgi	200	xhr	jquery-1.8.3	202 B	46 ms		
11 requests   20.8 KB transferred							
Console Animations							
Synchronous XMLHttpRequest or detrimental effects to the er >	n the mai nd user's	n threa	d is depreca ence. For mo	ted because of re help, check	its https:	iquery-1.8. //xhr.spec.whatwo	<u>3.min.js:</u> g.org/.

- 3. Select XHR to filter on the Ajax calls.
- 4. Execute the API call.

## Example

You want to perform the call <code>asset.get\_by\_url(url)</code> on an asset with this URL: cloudflow:// PP\_FILE\_STORE/notes.pdf.

In the JavaScript console the call is the following: api.asset.get\_by\_url("cloudflow://
PP\_FILE\_STORE/notes.pdf");

5. Open the request in the network panel to see the details.

× Heade	ers Preview Response Cookies Timing
▼ General Reque Reque Statu	sst URL: http://localhost:9090/portal.cgi est Method: POST & Code: © 200 OK
<ul> <li>Respon</li> <li>Acces</li> <li>Cache</li> <li>Conne</li> <li>Conte</li> <li>Expire</li> <li>Pragn</li> </ul>	se Headers view source ss-Control-Allow-Origin: * a-Control: no-cache, no-store, must-revalidate section: close ent-Type: application/json s: 0 na: no-cache
Reques Accep Accep Accep Accep Coch Conte Conte Cooki B878E Host: Origin Pragn Refer User- cko) X-Ref	<pre>tHeaders view source &gt;tt */* t-Encoding: gzip, deflate &gt;t-Encoding: gzip, deflate</pre>
▼ Form Da	<pre>ta view parsed hod":"asset.get_by_url","url":"cloudflow://PP_FILE_STORE/notes.pdf"}</pre>

- In the General section, you can see that the request was sent by a POST call to http://localhost:9090/ portal.cgi.
- In the **Form Data** section, you can see the submitted data. Make sure that you show the source version by clicking the **view source** button. The following data was sent:

```
{"method":"asset.get by url","url":"cloudflow://PP FILE STORE/notes.pdf"}
```

6. Now all the information needed is available to perform the call with cURL:

```
curl 'http://localhost:9090/portal.cgi'
--data '{"method":"asset.get_by_url","url":"cloudflow://PP_FILE_STORE/
notes.pdf"}'
```

#### 2. Pass the Authentication Information

However, when you do this you will get following error:

```
{
    "error_code":"no_permission",
    "error":"You don't have the correct permissions for the operation
    'asset.get_by_url' !"
}
```

The reason for this is that browser cookies are not sent. Therefore you are not authenticated and the request returns an error.

**Note:** Sending the browser cookies will help in this case but it is not convenient.

In this case, the best way to authenticate is using the session field of the method request. When this field contains a valid session key, it will use this as authentication information. All calls support this extra field.

To do this, follow these steps:

1. Create a session key with the following API call:

api.auth.create\_session(user\_name, session).

On the JavaScript console this would be:

api.auth.create\_session('admin');

2. You receive the following response:

```
{session:
"56e172b93d570000000000c0AA139F5D392075872DA1872949739EF1473716357"}
```

**3.** Add the session key to the previous cURL call:

4. You receive the asset record:

```
" id":"56e1819b4457000000001d7",
 "url":"PP FILE STORE/notes.pdf",
 "sub":"",
 "cloudflow": {
   "file":"cloudflow://PP FILE STORE/notes.pdf",
   "enclosing folder":"cloudflow://PP FILE STORE/",
   "part":"cloudflow://PP FILE STORE/notes.pdf"
},
"file name": "notes.pdf",
"file extension":".pdf",
"document name": "notes",
"path":["PP FILE STORE"],
"filetype": "application/pdf",
"modtime":1457517078,
"file size":3639539,
. . .
```

# CFApp

CFApp is a command line tool to install and download CLOUDFLOW applications.

CFApps are packages that contain files, folders, workflows and settings from a CLOUDFLOW system that work together. You can download and deploy this content on another CLOUDFLOW system to create an application running in CLOUDFLOW with an identical content.

# Install CFApp

1. Install node.js.

Node.js is a platform to write and run JavaScript applications. Since CFApp is written in node.js, it is necessary to install it.

 Download CFApp from https://www.npmjs.com/package/cfapp. Use this command:

```
sudo npm install -g cfapp
```

Now the CFApp command is available in the command line tool.

# Use CFApp

CFApp works with CFApp commands in the command line tool.

Some example commands:

• Command to print the help text:

CFApp

• Command to create a default project.CFApp file in the current directory:

CFApp app init

• Command to download a remote application:

CFApp app download /path/to/the/download/location/

You can also use command line options:

• Command line option to show the help of the app download command:

CFApp app download --help

• Command line option to provide the host and login data:

```
CFApp app download --host <hostname> --login <username> --password <password>
```

# Use case

This is an example of a use case of CFApp.

Wernham Hogg Labels Company is a label printer with their own development team. They use CLOUDFLOW to automate their workflows.

The development team has created a customized portal on top of CLOUDFLOW for their operators to manage the work. The portal is built with HTML, JavaScript and CSS files, stored in a File store. The File store also contains files and workflows.

They use CFApp for two reasons:

Compile data

With CFApp they can download all these files, folders and workflows and store them in a folder structure. Whenever they need to rebuild their portal on another CLOUDFLOW system, they use CFApp to upload the files, folders and workflows and make an exact copy of their running system on a different host.

• Updates

Every week they update the existing files, folders and workflows with an updated version.

This is their procedure:

#### 1. Create the folder

They have created a folder MyCFApp. This folder will contain the most recent version of CFApp.

## 2. Create the CFApp file

In the command line tool they enter the following command:

CFApp app init /path/to/the/MyCFApp/folder

- 1. They enter a name for the CFApp. If nothing is entered, the name will be the same as the folder name.
- 2. They enter a version number.

The version numbers are based on **Semantic Versioning 2.0.0**. This means that version numbers contain 3 numbers: **MAJOR.MINOR.PATCH**.

The very first CFApp will be version **0.0.1**. If nothing is entered, the system will by default use version 0.0.1.

The command will create the CFApp file in folder MyCFApp. This file is a JSON file which looks like this:

```
{
    "name": "MyCFApp",
    "version": "0.0.1",
    "files": [],
    "workflows": []
}
```

#### 3. Add files, folders and workflows

In the CFApp file they add which files, folders and workflows they want to download. To do this, they open the CFApp file in a text editor and edit it.

Example

```
{
    "name": "MyCFApp",
    "version": "0.0.1",
    "files": ["cloudflow://PP_FILE_STORE/Assets/"],
    "workflows": ["Jobs", "Copy Files"]
}
```

In this example, all files in folder **PP\_FILE\_STORE** > **Assets** will be downloaded, together with two workflows, **Jobs** and **Copy Files**.

#### 3. Download the files, folders and workflows

With this command they download the files, folders and workflows to the folder MyCFApp:

```
CFApp app download /path/to/the/MyCFApp/folder --host <hostname> --login <username> --password <password>
```

Where:

- Hostname is the name of the host where the files, folders and workflows are downloaded from.
- Username is the username to log in to CLOUDFLOW on the host where the files, folders and workflows are downloaded from.
- **Password** is the password to log in to CLOUDFLOW on the host where the files, folders and workflows are downloaded from.

Consequently, the folder structure will contain the following:
Nam	e
•	CFapps
	1708
	1709
	1710

- All the files and folders in **PP\_FILE\_STORE** > Assets, as indicated in the CFApp file.
- All the workflows indicated in the CFApp file.
- The CFApp file.

## 4. Upload the files, folders and workflows

With this command they upload the files, folders and workflows to a different host:

```
CFApp app upload /path/to/the/MyCFApp/folder --host <hostname> --login <username> --password <password>
```

Where:

- Hostname is the name of the host where the files, folders and workflows are uploaded to.
- Username is the username to log in to CLOUDFLOW on the host where the files, folders and workflows are uploaded to.
- **Password** is the password to log in to CLOUDFLOW on the host where the files, folders and workflows are uploaded to.

## 5. Updates

Every month they make a new version of the CFApp file with an updated version of the files, folders and workflows. In order to not lose data, they first commit the current files, folders and workflows in the MyCFApp folder to a version control system.

After this, they proceed as follows:

1. They download the updated files, folders and workflows with this command:

```
CFApp app download /path/to/the/MyCFApp/folder --host <hostname> --login <username> --password <password> --overwrite
```

Where:

- Hostname is the name of the host where the files, folders and workflows are downloaded to.
- Username is the username to log in to CLOUDFLOW on the host where the files, folders and workflows are downloaded to.
- **Password** is the password to log in to CLOUDFLOW on the host where the files, folders and workflows are downloaded to.
- Option --overwrite makes sure that former existing files, folders and workflows with the same name are overwritten with the newer versions. If this command option is not used, existing files, folders and workflows with the same name will be skipped.
- 2. They open CFApp file in an editor and change the version number:

```
{
    "name": "MyCFApp",
    "version": "0.0.2",
    "files": ["cloudflow://PP_FILE_STORE/Assets/"],
    "workflows": ["Jobs", "Copy Files"]
}
```

3. They update the existing files, folders and workflows in the folder MyCFApp with this command:

```
CFApp app update /path/to/the/MyCFApp/folder --host <hostname> --login <username> --password <password>
```

Where:

- Hostname is the name of the host where the files, folders and workflows are updated.
- Username is the username to log in to CLOUDFLOW on the host where the files, folders and workflows are updated.
- **Password** is the password to log in to CLOUDFLOW on the host where the files, folders and workflows are updated.

**Note:** Command **update** is only possible in case the version number is higher than the previous one. If you want to update with a version number that is equal to or lower than the previous one, you need to use the command option **--force**.

# MARS

MARS (Managed Application Repository Service) is a centralized repository for MARS apps.

# What is a MARS app

A MARS app is a package that contains assets, workflows, a UI and settings from a CLOUDFLOW system.

You can download and install this package on another CLOUDFLOW system and run the MARS app with identical content.

Parts of a MARS app:

- Assets: files and folders.
- Workflows: contains settings and optionally forms and scripts.
- HTML UI: a web site that contains HTML, JavaScript and CSS.
- Configuration:
  - Start-up settings for the server (users, scopes, etc...).
  - Database initial objects.
- **PAGEBUILDER**: Pagebuilder UI pages.

#### Types of MARS apps in MARS

There are three types of MARS apps in MARS:

- Public Apps: Public Apps are available for all MARS users. Public Apps have been tested extensively and comply to predefined requirements.
- My Apps: My Apps are MARS apps of which you are the Maintainer. You are a Maintainer if you have created a MARS app from scratch in MARS, or copied an existing MARS app in MARS in order to edit it. See User roles on page 400 for more information on MARS user roles.
- Apps Shared With Me: Apps Shared With Me are MARS apps that someone else has shared with you. You can install and use/review these MARS apps.

# **MARS Creator license**

You can use MARS without a MARS Creator license or with a MARS Creator license.

The way of using MARS is very different in both cases.

If you use MARS without a MARS Creator license, you cannot create your own MARS apps. You cannot
manage the MARS server yourself either. You can only specify to view or to hide the MARS UI in SETTINGS >
SETTINGS > MARS > SHOW MARS UI.

See Use MARS without a MARS Creator license on page 399 for a detailed overview of all the possibilities and functionalities of MARS if you use MARS without a Creator license.

If you use MARS with a MARS Creator license, you can create MARS apps yourself. You can also manage your own MARS server. In SETTINGS > SETTINGS > MARS > SERVER URL, you can specify the URL of the MARS server.

See Use MARS with a MARS Creator license on page 400 for a detailed overview of all the possibilities and functionalities of MARS if you use MARS with a Creator license.

# Use MARS without a MARS Creator license

If you use MARS without a MARS Creator license, you can access several MARS apps, but you cannot create MARS apps yourself or manage your own MARS server.

You have a list of MARS apps that you can install and use on your local system. If a new version of one of your installed MARS apps is available, you can update this MARS app.

## Show MARS user interface

To view MARS in the CLOUDFLOW user interface, you have to activate it.

To show the MARS user interface, enable the checkbox next to SETTINGS > SETTINGS > MARS > SHOW MARS UI.

## Your list of MARS apps

You have access to the following MARS apps:

## The MARS apps that you have a license for

These are all the MARS apps you have purchased.

You can **install** these MARS apps on your local system and use them. If there is a new version of a MARS app that you have installed available, you can **update** it. You cannot edit these MARS apps.

## The MARS apps that you co-own

These are the MARS apps that someone (typically a Hybrid Software technician) has created for you and that you coown.

You can **install** these MARS apps on your local system and use them. If needed, a Hybrid Software technician can **edit** these MARS apps and **upload** a new version to your system.

## The MARS apps that are shared with you

These are the MARS apps that someone (typically a Hybrid Software technician) has created for you and that are shared with you.

You can **install** these MARS apps on your local system and use them. If there is a new version of a MARS app that is shared with you available, you can **update** it. Contrary to MARS apps that you co-own, shared MARS apps cannot be edited.

## Installing and updating a MARS app

How to install and update MARS apps?

## Install a MARS app

When you want to use a MARS app, you have to install it on your local system. To do so, follow these steps:

- 1. Go to the MARS app.
- 2. Select Install.

The MARS app is now installed on your local system and you can use it.

## Update a MARS app

When there is a new version of one of your installed MARS apps available on MARS, you can update your local app. To do so, follow these steps:

- 1. Go to the installed MARS app.
- 2. Select Update.

# Use MARS with a MARS Creator license

## **User roles**

There are two different user roles in MARS.

## Maintainer

You are a Maintainer if you have:

- Created a MARS app from scratch in MARS.
- Copied an existing MARS app in MARS in order to edit it.

As a Maintainer, you can:

- Upload your newly created MARS app to MARS.
- Change the version of your MARS app and upload it.
- Request to publish your MARS app on MARS so that it becomes publicly available for other users.
- Share your MARS app with another user.
- Assign a **co-owner** to a MARS app. This is typically a customer who you have created a MARS app for and on which system you want to install the app. If a customer co-owns a MARS app, the app will be visible in the customer's list of MARS apps and you can install it on the customer's local system.

As a Maintainer, you are **responsible** for your MARS app. This implies that you need to:

- Maintain the MARS app.
- Guarantee the proper functioning of the MARS app.
- Provide and maintain clear and correct documentation for the MARS app.
- Provide feedback regarding the MARS app.

In default cases, the person who creates the MARS app is the Maintainer and is responsible for the MARS app. However, the responsibility of a MARS app can switch to another user when needed.

There are two levels of Maintainer:

- **Maintainer**: if you are a Maintainer, you are able to perform all actions in MARS by using the tools that are provided in the MARS User Interface.
- **Expert Maintainer**: if you are an expert Maintainer will use MARS to control a source control client by using functionalities that are hidden from the standard User Interface.

All Expert Maintainers are Maintainers. Not all Maintainers are Expert Maintainers.

## User

As a User, you can :

- Browse MARS apps created by Maintainers in MARS.
- **Install** MARS apps created by Maintainers in MARS. In this case, the MARS app is pulled from MARS and installed on the User's CLOUDFLOW system, without editing the MARS app. The link between the MARS app on MARS and the MARS app installed on the User's system remains. If the Maintainer updates the MARS app, these updates can also be installed on the MARS app that was installed on the User's system by using an update functionality.
  - () Important: Users must not edit the MARS apps they install.
- **Copy** MARS apps created by Maintainers in MARS. In this case, the MARS app is copied to the User's CLOUDFLOW system, with the purpose of editing the MARS app. The link between the MARS app on MARS and the MARS app copied by the user is broken. If the Maintainer updates the MARS app, the User will not be able to install these updates.



**Important:** If a User copies a MARS app to his own CLOUDFLOW system, the User becomes the Maintainer.

All Maintainers are Users. Not all Users are Maintainers.

## **Types of MARS apps**

There are three types of MARS apps in MARS.

There are:

- Public Apps
- My Apps
- Apps Shared With Me

Each type contains available and installed MARS apps.

- Available means that the MARS apps are available on the MARS server.
- Installed means that the MARS apps are installed on your local CLOUDFLOW system.

In the User Interface, there is a tab for each type of MARS app.

**Important:** MARS Apps created for demo purposes are watermarked. These Apps must never be used in a production environment.

## **Public Apps**

This tab contains an overview of all the public MARS app.

## Available

MARS apps in **Public Apps** > **Available** are MARS apps that can be accessed by everyone who has a CLOUDFLOW installation with a MARS license. These MARS apps must comply with a set of predefined quality requirements. See MARS app design guidelines on page 409 for more information.

The following action buttons are available in **Public Apps** > **Available**:

• **Install**: to install the MARS app on your local system. The link between the original public MARS app and the installed MARS app remains, which means that the installed MARS app can be updated when a new version of the original public MARS app is published.

- **Details**: to display the details of the MARS app.
  - **Description**: the description of the MARS app. This description is entered by the Maintainer when creating the MARS app.
  - Release Notes: the release notes of the MARS app.
  - **Documentation**: the documentation of the MARS app.
    - **Note:** The Maintainer is required to provide clear and extended documentation in case of a Public MARS app.
  - Maintainer: the Maintainer of the MARS app. See User roles on page 400 for more information.
  - Versions: the existing versions of the MARS app.
- **Copy to My Apps**: to copy the MARS app to **My Apps** > **Installed**. The link between the original public MARS app and the installed MARS app is broken. If you copy a MARS app, you become the Maintainer of the MARS app.

## Installed

MARS apps in **Public Apps** > **Installed** are MARS apps that you have installed from the MARS apps in **Public Apps** > **Available**.

**Note:** Installed public MARS apps should never be edited. If an installed public MARS app is edited, it will be impossible to update it to the latest version of the MARS app that is publicly available.

The following action buttons are available in **Public Apps > Installed**:

- Delete locally: to delete the installed public MARS app from your local system. If you select the checkbox Remove local files and workflows, the files and folders on your local system are also removed.
- **Details**: to display the details of the MARS app.
  - **Description**: the description of the MARS app. This description is entered by the Maintainer when creating the MARS app.
  - **Release Notes**: the release notes of the MARS app.
  - **Documentation**: the documentation of the MARS app.
    - **Note:** The Maintainer is required to provide clear and extended documentation in case of a Public MARS app.
  - Maintainer: the Maintainer of the MARS app. See User roles on page 400 for more information.
  - Versions: the existing versions of the MARS app.
- Update: to update the installed public MARS app to the latest version of the MARS app in Public Apps > Available.

#### My Apps

This tab contains an overview of all the MARS apps of which you are the Maintainer.

## Available

MARS apps in My Apps > Available are available on your personal space on MARS and are only accessible for you.

The following action buttons are available in **My Apps** > **Available**:

• Install: to install the MARS app on your local system.

- **Details**: to display the details of the MARS app.
  - **Description**: the description of the MARS app. This description is entered by the Maintainer when creating the MARS app.
  - Release Notes: the release notes of the MARS app.
  - **Documentation**: the documentation of the MARS app.

**Note:** The Maintainer is required to provide clear and extended documentation in case of a Public MARS app.

- Maintainer: the Maintainer of the MARS app. See User roles on page 400 for more information.
- **Co-owners**: the co-owners of the MAPS app. Co-ownership makes it possible for several Maintainers to collaborate on the same MARS app. All co-owners can copy and edit a MARS app, and upload the changes to MARS.
- This is typically a customer who you have created a MARS app for and on which system you want to install the app. If a customer co-owns a MARS app, the app will be visible in the customer's list of MARS apps and you can install it on the customer's local system.
- Shared With: to share the MARS app with other users and to view the Users with whom the MARS app is shared. See Share a MARS app with another User on page 408 for more information.
- Copy Of: the original MARS app, in case the selected MARS app is a copy.
- Versions: the existing versions of the MARS app.
- Copy: to copy the MARS app to your personal space on MARS.
- Delete from MARS: to delete the MARS app from your personal space on MARS.
- **Request to publish**: to send a mail with a request to publish the MARS app. Only if the MARS app complies with the predefined quality requirements, the MARS app is published and will be displayed in **Public Apps** > **Available**. See MARS app design guidelines on page 409 for more information on the requirements for Public Apps.

#### Installed

MARS apps in **My Apps** > **Installed** are MARS apps that are installed on your local system. Newly created MARS apps are installed here.

The following action buttons are available in My Apps > Installed:

- Delete locally: to delete the installed MARS app from your local system. If you select the checkbox **Remove local** files and workflows, the files and folders on your local system are also removed.
- Upload to MARS: to upload the (edited) MARS app to your personal space on MARS.
- Edit: to edit the MARS app. See Edit a MARS app on page 405 for more information on how to edit a MARS app.

#### Apps Shared With Me

This tab contains an overview of all the MARS apps that a Maintainer has shared with you.

## Available

MARS apps in **My Apps** > **Available** are MARS apps that are maintained by the Maintainer and that are shared with you. You can install the MARS app that are shared with you for example to review them, or you can copy them to edit them.

The following action buttons are available in **My Apps** > **Available**:

- Install: to install the MARS app on your local system.
- Copy to My Apps: to copy the MARS app to My Apps > Available. The link between the original shared MARS app and the copied MARS app is broken. If you copy a MARS app, you become the Maintainer of the MARS app.

- **Details**: to display the details of the MARS app.
  - **Description**: the description of the MARS app. This description is entered by the Maintainer when creating the MARS app.
  - **Release Notes**: the release notes of the MARS app.
  - **Documentation**: the documentation of the MARS app.
    - **Note:** The Maintainer is required to provide clear and extended documentation in case of a Public MARS app.
  - Maintainer: the Maintainer of the MARS app. See User roles on page 400 for more information.
  - Versions: the existing versions of the MARS app.

#### Installed

MARS apps in **My Apps** > **Installed** are MARS apps that are shared with you and that you have installed on your local system.

The following action buttons are available in **My Apps** > **Installed**:

- Delete locally: to delete the MARS app from your local system. If you select the checkbox **Remove local files** and workflows, the files and folders on your local system are also removed.
- Update: to update the MARS app to the latest version of the MARS app in My Apps > Available of the first user. The first user is the user who has shared the MARS app with you.
- **Details**: to display the details of the MARS app.
  - **Description**: the description of the MARS app. This description is entered by the Maintainer when creating the MARS app.
  - **Release Notes**: the release notes of the MARS app.
  - **Documentation**: the documentation of the MARS app.
    - **Note:** The Maintainer is required to provide clear and extended documentation in case of a Public MARS app.
  - Maintainer: the Maintainer of the MARS app. See User roles on page 400 for more information.
  - Versions: the existing versions of the MARS app.

## How to...

#### Install a Public MARS app

This chapter describes the steps needed to install a Public MARS app.

To install a MARS app in **Public Apps** > **Available**, follow these steps:

- 1. Go to the MARS app in **Public Apps** > Available.
- 2. Select Install.

The MARS app is now installed on your local system and is displayed in Public Apps > Installed.

Attention: Keep in mind not to edit the MARS app. If you do edit the MARS app, you will not be able to update the MARS app in case a newer version becomes available in **Public Apps** > Available.

#### Update an installed Public MARS app

This chapter describes the steps needed to update an installed Public MARS app if a new version is available.

To update an installed public MARS app in **Public Apps** > **Installed**, follow these steps:

- 1. Go to the MARS app in **Public Apps** > **Installed**.
- 2. Select Update.

The MARS app is now updated to the version on **Public Apps** > **Available**.

**Note:** This is only possible if you have not edited the MARS app.

## Copy a Public MARS app

This chapter describes the steps needed to copy a Public MARS app.

To copy a MARS app in **Public Apps** > **Available**, follow these steps:

- 1. Go to the MARS app in **Public Apps** > Available.
- 2. Select Copy.

The MARS app is now copied to your local system and is displayed in **My Apps** > **Installed**. This means that from now on, you are the Maintainer of the MARS app and you have certain responsibilities. See Maintainer for more information on the Maintainer's responsibilities for the MARS app.

## Create a new MARS app

This chapter describes the steps needed to create a new MARS app.

To create a new MARS app, follow these steps:

- 1. Open MARS.
- 2. Select New....
- 3. Enter a Name and a Description for your MARS app.
- 4. Select Create New.

The empty MARS app is now stored in the list **My Apps** > **Installed**.

## Edit a MARS app

This chapter describes the steps needed to edit a MARS app.

A MARS app contains:

- Parameters (a version number, a description, documentation...)
- Content (workflow and assets)

To add or to change the parameters and/or content of a MARS app of which you are the Maintainer, follow these steps:

- 1. Go to the MARS app in **My Apps** > **Installed**.
- 2. Select Edit.

- 3. Enter or edit the parameters:
  - The icon on the top right: here you can change the icon of the MARS app. Hover over the icon, click to select an asset to use as icon and select **Choose**.
  - **Description**: here you can enter a description for the MARS app. The description is used in MARS by users to refer to the MARS app and is displayed next to the icon in the MARS app list.
  - Version: here you can enter a version for the MARS app. You can enter a version in the version field, or you can use the arrows to change the Major, Minor and Patch version.
  - **Documentation Path**: here you can define the path where the documentation file of the MARS app is located. See Add documentation and release notes to a MARS app on page 406 for more information on how to add documentation to a MARS app.
  - Assets: here you can select assets to add to the MARS app.
    - a. Select Add Asset....
    - **b.** Select the asset.
    - c. Select Choose.
    - d. If needed, repeat to select another asset.

**Note:** Make sure that the assets of the MARS app are located in a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app.

- Workflows: here you can select workflows to add to the MARS app.
  - **a.** Select the workflow from the drop-down list.
- b. Select Add Workflow.
- c. If needed, repeat to select another workflow.

## 4. Select Save.

The result is a MARS app in **MARS** > **My Apps** > **Installed**. The MARS app is installed on your local system. Since you have created the MARS app, you are the Maintainer.

#### Change the icon of a MARS app

This chapter describes the steps needed to change the icon of your MARS app.

All MARS apps have an icon. If you don't change the icon of your MARS app, the default icon is displayed in the list:

CERM integration kit

However, to make your MARS app more recognizable, you can change the default icon:



Transfer large files with CloudflowTransfer app

To change the icon of your MARS app, follow these steps:

- 1. Go to the MARS app in **My Apps** > **Installed**.
- 2. Select Edit.
- 3. Hover over the MARS app icon on the top right and click Select.
- 4. Choose a CLOUDFLOW asset to use as icon for your MARS app.
- 5. Select Save at the bottom.

#### Add documentation and release notes to a MARS app

This chapter describes the steps needed to add or change the documentation or release notes of a MARS app.

#### Add documentation

To add documentation to a MARS app, follow these steps:

- 1. Make sure there is a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app.
- 2. Create a Markdown file **readme.md** and save it in the subfolder. For a cheatsheet about the Markdown format, refer to https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet.
- **3.** Add documentation content to the markdown file. Make sure that the documentation is clear and complete. See Documentation Guidelines for more information on documentation guidelines.
- 4. Go to the MARS app in **My Apps** > **Installed**.
- 5. Select Edit.
- 6. Specify the documentation path in **Documentation Path**.

## Edit the Markdown file

If the MARS App contains a Pagebuilder file, you can open and edit the Markdown file in PAGEBUILDER. To do so:

- 1. Open a Pagebuilder file that is part of the MARS App.
- 2. On the tab Cfapp on the left side.
- **3.** Select the readme.md file.
- 4. Edit the readme.md file. You can see the result while editing.

## Add release notes

To add release notes to a MARS app, follow these steps:

- 1. Make sure there is a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app.
- 2. Create a Markdown file changelog.md and save it in the subfolder.
- **3.** Add content to the markdown file.
- 4. Go to the MARS app in **My Apps** > **Installed**.
- 5. Select Edit.
- 6. Specify the path of the release notes in **Documentation Path**.

#### Upload a MARS app to MARS

This chapter describes the steps needed to upload a MARS app to MARS.

To upload a MARS app to MARS, follow these steps:

- 1. Go to the MARS app in My Apps > Installed.
- 2. Select Upload to MARS.
  - **Note:** If you select the checkbox **force push**, you can force the application to upload in case the MARS application is out of sync with the server, and put your version as the latest version on the server.

The MARS app is now uploaded to MARS. Only you can access it.

#### Change the version of a MARS app and upload it

This chapter describes how to change the version of your MARS app and upload it to MARS.

You can edit existing MARS app in **My Apps** > **Installed** at all times.

Follow these steps:

- 1. Go to the MARS app in **My Apps** > **Installed**.
- 2. Select Edit.
- **3.** If needed, change the **Version** number of the CLOUDFLOW Application. See Change the Version Number of a MARS app for more information on the guidelines.
- 4. Make the needed edits. For example, add a workflow.
- 5. Select Save.
- 6. Select Upload to MARS to upload the edited version of the MARS app.

The existing MARS app on **My Apps** > **Available** is now replaced by the current version of the MARS app in **My Apps** > **Installed**.

- **Note:** You can only edit MARS apps that are installed on your local system, in **My Apps > Installed**. If you want to edit a MARS app in **My Apps > Available**, you need to install it first.
- **Note:** In case the MARS app exists in both **My Apps** > **Available** and **My Apps** > **Installed**, it will not be possible to install the MARS app from **My Apps** > **Available** to **My Apps** > **Installed**. If you want to do this, you need to delete the local CLOUDFLOW Application in **My Apps** > **Installed**.

## Send a request to publish a MARS app

This chapter describes the steps needed to send a request to publish your MARS app.

If you want to make a MARS app on **My Apps** > **Available** publicly available, it needs to be published on **Public Apps** > **Available**.

Attention: Only MARS apps that comply with predefined requirements specified here are published.

To send a request to publish a MARS app on **My Apps** > **Available**, follow these steps:

- 1. Go to the MARS app in My Apps > Available.
- 2. Select Request to publish.
- **3.** Your mail client opens and displays a template mail. The mail contains information about the MARS app and a publish request.

Only if the MARS app complies with all requirements and it has passed the QA cycles, it will be made publicly available in **Public Apps** > **Available** > **.** 

## Share a MARS app with another User

This chapter describes the steps needed to share your MARS app with another user.

You can only share your Available MARS apps, these are the MARS apps that are uploaded to MARS.

To share a MARS app of which you are the Maintainer with another user, follow these steps:

- 1. Go to the MARS app in My Apps > Available.
- 2. Select Details.
- 3. Go to Shared With and select Share With.
- 4. Enter the customer code of User with whom you want to share your MARS app. You can find the customer code in **SETTINGS** > **LICENSE**.

When you have shared a MARS app with a User, this User will see the MARS app in Apps Shared With Me > Available. The User can do the following:

- Install the MARS app without editing it. In this case the MARS app is installed on the User's local system and is displayed in Apps Shared With Me > Installed.
- Copy the MARS app to edit it. In this case the MARS app is copied to the User's local system and is displayed in My Apps > Installed. The User is now the Maintainer of the MARS app.

## Assign a co-owner

This chapter describes the steps needed to assign a co-owner of your MARS app.

Co-ownership of a MARS App makes it possible to create a MARS app for another user, and co-own the MARS App together. This means that all co-owners can install the MARS App, edit it and upload the edits. For case of Hybrid Software technicians, a typical use case is to create a MARS App for a customer, and add the customer as a co-owner.

However, collaboration on a MARS App needs to be applied very carefully. If two maintainers collaborate on the same MARS app, they can only upload edits in case the installed MARS app on both Maintainers' local systems is identical. It is not possible to merge edits. Consequently, collaborators need to coordinate the changes in a MARS app and the upload very well.

To assign a co-owner to a MARS app, follow these steps:

- 1. Go to the MARS app in My Apps > Available.
- 2. Select Details.
- 3. Go to Co-owners and select Add Owner ....

4. Enter the customer code of the User who you want to assign as a co-owner.

## MARS app design guidelines

MARS apps should be of high quality so a smooth installation and problem-free use of the MARS app is guaranteed.

## Name

All MARS apps must contain a Name. The MARS app Name is a unique identifier for the MARS app. It is used in for example URLs and directories.

A MARS app Name:

- Must be unique.
- Must only contain small caps.
- Must only contain alphanumeric characters.

## Example

cerm

## **Descriptive Name**

All MARS apps must contain a Descriptive Name. The MARS app Descriptive Name is the name that is used in MARS by users to refer to the MARS app. The Descriptive Name should contain a clear description of the MARS app.

## Example

Cerm Integration Kit

## Version number

All MARS apps must contain a version number. The Version number follows the convention of Semantic Versioning, described on https://semver.org/.

These are the guidelines for incrementing the Major, Minor or Patch version number:

- Increment the Major version when you make incompatible API changes.
- Increment the Minor version when you add functionality in a backwards-compatible manner.
- · Increment the Patch version when you make backwards-compatible bug fixes.

## Example

- 1.0.0 for the first version of a MARS app.
- 1.1.0 when you have added some new functionality.
- 1.2.0 when you have added some other new functionality.
- 1.2.1 when you have fixed a bug in the MARS app.
- 2.0.0 when you created a new major version of the MARS app and the code of the MARS app becomes backwards incompatible

## Files<sup>\*</sup>

Files must be added to a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app.

## Example

PP\_FILE\_STORE/cerm/...

## HTML pages

If the MARS app is a Template Application, HTML pages must be built with PAGEBUILDER.

## Workflows

The names of the workflows must start with the MARS app Name.

#### Example

cerm-xxx

## Documentation

All MARS apps must contain documentation. The documentation file is a Markdown that must be added to a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app.

The documentation must be clear and extended so the use and purpose of the MARS app is evident for all Users.

For Public MARS apps, the provided documentation is used to test the MARS app.

## General

Create your MARS app preferably in such a way that it does not need any modification. If this is not possible, make sure that the MARS app needs as less modification as possible.

## KISS (Keep It Simple, Stupid)

<sup>\*</sup> These guidelines are **enforced** for Public MARS apps. This means that all MARS apps that are requested to be published will only be published in case they comply with all these guidelines.

For all other MARS apps, these guidelines are strongly recommended.

# **Scenarios**

This chapter describes some example scenarios where the use of MARS is involved.

## **Parties involved**

The following parties play a part in one of the scenarios below.

- Future Software is a software company that develops customized software systems for customers in the packaging industry. Future Software supplies services for both **Dunder Mifflin Prepress**, a company that provides prepress services for label printers and **Wernham Hogg Labels**, a label printer.
- **Rick** is a support engineer who works for Future Software. He helps customers when they have questions or problems. One of his main tasks is setting up customized workflows and pages for customers. He mainly works in his office at Future Software and does not visit customers. Rick has a CLOUDFLOW license with customer code **BE-RICK**.
- **Maggie** is also a support engineer at Future Software. Her main task is to prepare and to go to the customer's site to do CLOUDFLOW installations and to setup customized workflows and HTML pages. Maggie has a CLOUDFLOW license with customer code **BE-MAGGIE**.
- **Daryl** is a software engineer who also works for Future Software. Daryl writes and edits the software tools that are developed by Future Software. He does not have any contact with customers. Daryl has a CLOUDFLOW license with customer code **BE-DARYL**.

## **Dunder Mifflin Prepress**

- **Dunder Mifflin Prepress** is a company that provides prepress services for label printers. Dunder Mifflin Prepress is a customer of **Future Software** and supplies services for **Wernham Hogg Labels**, a label printer. Dunder Mifflin Prepress has a CLOUDFLOW license with customer code **BE-DMPPR**.
- Judith is a project manager at Dunder Mifflin Prepress. She uses CLOUDFLOW to submit jobs via KIOSK. Her experience with creating and maintaining workflows in CLOUDFLOW is limited.
- **Glenn** is an IT responsible at Dunder Mifflin Prepress. He does not use CLOUDFLOW himself for his job but he knows very well how to create, edit and maintain all the CLOUDFLOW workflows at Wernham Hogg Prepress.

## Wernham Hogg Labels

- Wernham Hogg Labels is a label printer. They design, prepare and print labels for the packaging industry. Wernham Hogg Labels is a customer of both Future Software and Dunder Mifflin Prepress. Wernham Hogg Labels has a CLOUDFLOW license with customer code BE-WHL.
- **Carl** is a prepress operator at Wernham Hogg Labels. His job is to prepare for print the labels that were designed by her colleague designers in PACKZ. When the label is ready for print, he saves the label in the CLOUDFLOW File store. Carl does not have any knowledge of CLOUDFLOW.
- Lori is the prepress and CLOUDFLOW responsible at Wernham Hogg Labels. Among other things, she needs to approve all labels that were prepared by Carl. Only after her approval, the labels can be sent to the press. Lori only has knowledge of CLOUDFLOW and MARS.

## Scenario: installing a public MARS app

This scenario includes the context and approach for installing a MARS app that is publicly available on MARS.

## Context

**Rick**, a support engineer at **Future Software**, experiences some performance issues on his CLOUDFLOW system. He wants to test it using the MARS app **quickbenchmark**.

**quickbenchmark** is a MARS app that tests the performance of your CLOUDFLOW system. It is publicly available on MARS and can be installed by everyone who has a CLOUDFLOW system with a MARS license. The MARS app must not be edited after installation.

Rick has not yet installed quickbenchmark before, so he has to install the MARS app.

## Approach

Rick takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS > Public > Available. This list gives an overview of all the CLOUDFLOW Applications that are publicly available and can be installed.
- 3. Next to the MARS app quickbenchmark, he selects Install.

The MARS app is installed locally on Rick's system.

## Scenario: updating a public MARS app

This scenario includes the context and approach for updating a MARS app that was installed from the public space on MARS.

## Context

**Rick**, a support engineer at **Future Software** installed the public MARS app **quickbenchmark**, for testing the performance of his computer (see Scenario: installing a public MARS app on page 411). An updated version of **quickbenchmark** is now available on MARS. Since Rick uses the MARS app **quickbenchmark** on a regular basis, he wants to install the updated version. This means that he has to update his installed version of the MARS app.

## Approach

Rick takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS > Public > Installed. This list gives an overview of all the MARS app that have been installed locally in Rick's system.
- 3. Next to the MARS app quickbenchmark, he selects Update.

The MARS app **quickbenchmark** that is installed on Rick's local system is now updated to the latest version that is publicly available on MARS.

## Scenario: copying a public MARS app

This scenario includes the context and approach for copying a MARS app that is publicly available on MARS.

## Context

On MARS, the public MARS app **approvalportaldmf** is available. This MARS app was created by **Maggie**, a support engineer at **Future Software**. It contains some approval workflows and a customized HTML page for one of her customers, **Dunder Mifflin Prepress**. In combination with a customized HTML page for Dunder Mifflin Prepress, the workflows can be used as a portal. Since many content and settings of Maggie's MARS app can be reused for other customers as well, it has been published on the public space on MARS (see Scenario: publishing a MARS app on page 415).

**Rick**, Maggie's support engineer colleague at **Future Software** also wants to use **approvalportaldmf** for one of his own customers, **Wernham Hogg Labels**. He can reuse many of the content and settings of the existing MARS app, but some of its content is customized for Maggie's customer, Dunder Mifflin Prepress. However, Rick can use the existing MARS app as a template and customize it for Wernham Hogg Labels. For example, Rick has edited the existing HTML page in **approvalportaldmf** and customized it with Pagebuilder for Wernham Hogg Labels.

This means that Rick has to copy **approvalportaldmf** from the public space on MARS to his own system and edit it.

## Approach

First, Rick makes sure that the location of the assets, the workflow names and documentation comply to the guidelines regarding MARS apps. Although he is not planning to make the edited MARS app public, he has decided to follow the recommended guidelines. See MARS app design guidelines on page 409 for more information. He has been proactive and has decided that he will name the MARS app **approvalportalwhl**. Consequently:

- He has stored all the assets in a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app, so PP\_FILE\_STORE/approvalportalwhl/....
- He has created a Markdown file **readme.md** that contains clear and extensive documentation on the use and functionality of the MARS app. The path of the documentation file is PP\_FILE\_STORE/ approvalportalwhl/documentation/readme.md.
- All the workflow names start with the MARS app name, so approvalportalwhl-workflow-xxx .

Next, Rick copies the MARS app:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS > Public > Available. This list gives an overview of all the CLOUDFLOW Applications that are publicly available and can be copied.
- 3. Next to the MARS app approvalportaldmf, he selects Copy to My Apps.
- 4. He enters the new name for the MARS app: approvalportalwhl.
- 5. He selects Copy.

The MARS app **approvalportalwhl** is now copied to Rick's personal space on MARS. From now on, he is the Maintainer of the MARS app, which means that he is responsible for it. See User roles on page 400 for more information.

Next, Rick needs to edit the MARS app and add the correct content to it so it can be used for Wernham Hogg Labels. Therefore, he first needs to install the MARS app to his local system:

1. In MARS > My Apps > Available, she selects Install next to the MARS app approvalportalwhl.

The MARS app is now installed on Rick's local system and is in the list in MARS > My Apps > Installed. The MARS app does not contain the correct content yet.

He takes the following steps to edit the content:

- 1. In MARS > My Apps > Installed, he selects Edit next to the MARS app approvalportalwhl.
- 2. He selects the version number of the MARS app. Since this is the first version of the MARS app, Rick chooses 0.0.1.

- 3. He browses to the path where the documentation file is located: PP\_FILE\_STORE/approvalportalwhl/ documentation/.
- 4. He adds the correct assets (for example, the HTML file that he has customized using PAGEBUILDER), which are located in PP FILE STORE/approvalportalwhl/.
- **5.** He adds the workflows.
- 6. He saves.

The MARS app **approvalportalwhl** now has the correct and customized content, but it is not yet available on MARS.

Rick takes the following steps to upload the MARS app to MARS:

- 1. In MARS > My Apps > Installed, he selects Upload to MARS next to the MARS app approvalportalwhl.
- 2. He selects Push to MARS.

The MARS app is now uploaded to Rick's personal space on MARS.

## Scenario: creating and uploading a MARS app

This scenario includes the context and approach for creating a new MARS app and uploading it to MARS.

## Context

**Daryl**, a software engineer at **Future Software** has created some CLOUDFLOW workflows to test software builds. He wants to add them to a MARS app and upload the MARS app to MARS in order to conserve the MARS app and all its future versions.

Daryl does not have a MARS app on his system where he can add the workflows to, so he has to create a new MARS app, add the workflows and upload it to MARS.

## Approach

First, Daryl needs to create the MARS app.

He takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS.
- 3. He selects New....
- **4.** He enters the following parameters:
  - The Name of the MARS app: testmysoftwarebuilds.
  - The Description of the MARS app. He chooses MARS app to test my software builds.
- 5. He selects Create New.

The MARS app is now created locally on Daryl's system and is in the list in MARS > My Apps > Installed. The MARS app does not contain any content yet.

Next, Daryl needs to add the workflows to the MARS app.

- 1. In MARS > My Apps > Installed, he selects Edit next to the MARS app testmysoftwarebuilds.
- 2. He selects the version number of the MARS app. Since this is the first version of the MARS app, Daryl chooses 0.0.1.
- **3.** He adds the workflows.
- 4. He saves.

The MARS app now contains the workflows, but it is not yet uploaded to MARS.

Daryl takes the following steps to upload the MARS app to MARS:

- 1. In MARS > My Apps > Installed, he selects Upload to MARS next to the MARS app testmysoftwarebuilds.
- 2. He selects Push to MARS.

The MARS app **testmysoftwarebuilds** is now uploaded to Daryl's personal space on MARS. It is only available for Daryl, it is not publicly available. The MARS app exists in both **MARS** > **My Apps** > **Available** and **MARS** > **My Apps** > **Installed**. In this case, Daryl will not be able to install the MARS app from **MARS** > **My Apps** > **Available** to **MARS** > **My Apps** > **Installed**. If he wants to do that, he first needs to delete the MARS app on **MARS** > **My Apps** > **Installed**.

## Scenario: upload a new version of a MARS app to My Apps

This scenario includes the context and approach for uploading a new version of MARS app to MARS.

## Context

**Daryl**, a software engineer at **Future Software** has created **testmysoftwarebuilds**, a MARS app that contains some CLOUDFLOW workflows to test software builds. He has uploaded version 0.0.1 to MARS (see Scenario: creating and uploading a MARS app on page 413).

In the meantime, Daryl has fixed some bugs in the workflows in the MARS app. He does not want to lose this work, so he decides to upload the updated version of the MARS app to MARS.

## Approach

Daryl takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- He opens MARS > My Apps > Installed. Since Daryl has not deleted the MARS app from MARS > My Apps > Installed, it is still available. If Daryl had deleted the MARS app, he would have needed to reinstall it before he could edit it.
- 3. In MARS > My Apps > Installed, he selects Edit next to the MARS app testmysoftwarebuilds.
- **4.** He changes the version number of the MARS app. He fixed some bugs and the MARS app stays backwards compatible, so he increments the Patch number from 0.0.1 to 0.0.2.
- 5. He selects Save. As a consequence, the version of Daryl's MARS app testmysoftwarebuilds in MARS > My Apps > Installed and the version of the same MARS app in MARS > My Apps > Available is different. So it is very important that Daryl uploads the MARS app to MARS in order not to lose the bug fixes in the workflows.
- 6. In MARS > My Apps > Installed, he selects Upload to MARS next to the MARS app testmysoftwarebuilds.

**testmysoftwarebuilds** is now uploaded to MARS. Both the version of the MARS app in **MARS** > **My Apps** > **Available** and in **MARS** > **My Apps** > **Installed** are identical.

## Scenario: installing a MARS app from My Apps

This scenario includes the context and approach for installing a MARS app that is available on My Apps.

#### Context

**Daryl**, a software engineer at **Future Software** has created **testmysoftwarebuilds**, a MARS app that contains some CLOUDFLOW workflows to test software builds (see Scenario: creating and uploading a MARS app on page 413). He has uploaded the MARS app to MARS with current version 0.0.2 (see Scenario: upload a new version of a MARS app to My Apps on page 414. Because Daryl thought that the MARS app did not need any more updates, he deleted the MARS app from his system on **MARS** > **My Apps** > **Installed**. However, he has discovered another bug in one of the workflows and needs to update **testmysoftwarebuilds**.

Daryl deleted **testmysoftwarebuilds** on his local system, so he needs to reinstall the MARS app to be able to edit it. To make sure to not lose the edits to the MARS app, he needs to upload it to MARS.

#### Approach

Daryl takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. In MARS > My Apps > Available, he selects Install next to the MARS app testmysoftwarebuilds.

The MARS app is now installed in Daryl's local system and can be edited.

- 1. In MARS > My Apps > Installed, he selects Edit next to the MARS app testmysoftwarebuilds.
- 2. He changes the version number of the MARS app. Since he fixed some bugs and the MARS app stays backwards compatible, he increments the Patch number from 0.0.2 to 0.0.3.
- 3. He selects Save. As a consequence, the version of Daryl's MARS app in MARS > My Apps > Installed and the version of the same MARS app in MARS > My Apps > Available is different. So it is very important that Daryl uploads the MARS app to MARS in order not to lose the bug fixes in the workflows.
- 4. In MARS > My Apps > Installed, he selects Upload to MARS next to the MARS app testmysoftwarebuilds.

The MARS app is now uploaded to MARS. Both the version of the MARS app in MARS > My Apps > Available and in MARS > My Apps > Installed are identical.

## Scenario: publishing a MARS app

This scenario includes the context and approach for publishing a MARS app on MARS.

## Context

**Maggie**, a support engineer at **Future Software** has created some approval workflows for one of her customers, **Dunder Mifflin Prepress**. In combination with a customized HTML page for Dunder Mifflin Prepress, these workflows can be used as a portal. The workflows and the HTML page will be very useful for other CLOUDFLOW users as well, so Maggie wants to add the workflows and the HTML page to a MARS app and make it publicly available on MARS.

This means that the MARS app, including the approval workflows and the HTML page will be available for all CLOUDFLOW users that have a MARS license.

#### Approach

First, Maggie makes sure that the location of the assets, the workflow names and documentation comply to the guidelines regarding public MARS apps. See MARS app design guidelines on page 409 for more information. She has been proactive and has decided that she will name the MARS app **approvalportaldmf**. Consequently:

- She has stored all the assets in a subfolder in the root of PP\_FILE\_STORE with an identical name as the MARS app, so PP\_FILE\_STORE/approvalportaldmf/....
- She has created a Markdown file **readme.md** that contains clear and extensive documentation on the use and functionality of the MARS app. The path of the documentation file is PP\_FILE\_STORE/ approvalportaldmf/documentation/readme.md.
- All the workflow names start with the MARS app name, so approvalportaldmf-workflow-xxx .

Next, Maggie creates the MARS app:

- 1. She logs in to CLOUDFLOW with her username and password.
- 2. She opens MARS.
- 3. She selects New....
- 4. She enters the following parameters:
  - The Name of the MARS app: approvalportaldmf.
  - The Description of the MARS app. She chooses Approval Portal for Dunder Mifflin Prepress.
- 5. She selects Create New.

The MARS app is now created on Maggie's local system and is in the list in MARS > My Apps > Installed. The MARS app does not contain any content yet.

Maggie takes the following steps to add content:

- 1. In MARS > My Apps > Installed, she selects Edit next to the MARS app approvalportaldmf.
- 2. She selects the version number of the MARS app. Since this is the first version of the MARS app, Maggie chooses 0.0.1.

- 3. She browses to the path where the documentation file is located: PP\_FILE\_STORE/approvalportaldmf/ documentation/.
- 4. She adds the assets, which are located in PP FILE STORE/approvalportaldmf/.
- 5. She adds the workflows.
- **6.** She saves.

The MARS app **approvalportaldmf** now contains documentation, the assets and the workflows, but it is not yet available on MARS.

Maggie takes the following steps to upload the MARS app to MARS:

1. In MARS > My Apps > Installed, she selects Upload to MARS next to the MARS app approvalportaldmf.

2. She selects **Push to MARS**.

The MARS app is now uploaded to MARS, but because it is not publicly available, only Maggie can access it. To make it available for all MARS users, the application needs to be published on MARS.

Maggie takes the following steps to publish the MARS app so it becomes publicly available:

1. In MARS > My Apps > Available, she selects Request to publish next to the MARS app approvalportaldmf...

Maggie's mail client automatically opens a template mail that contains a **request to publish** message. The person who is responsible for the QA of the MARS app receives this message with the necessary information and makes sure the MARS app is tested against predefined MARS app design guidelines. See MARS app design guidelines on page 409 for more information. Only if the MARS app passes the QA tests, it will be published on MARS and made publicly available on MARS > Public Apps > Available.

## Scenario: sharing a MARS app

This scenario includes the context and approach for sharing a MARS app with another User.

## Context

Some time ago, **Rick**, a support engineer at **Future Software**, experienced some performance issues on his CLOUDFLOW system. To test it, he installed the public MARS app **quickbenchmark** (see Scenario: installing a public MARS app on page 411). Currently, one of his customers, **Wernham Hogg Labels**, also experiences performance issues on their CLOUDFLOW system. Rick wants to check what the problem is by using the MARS app **quickbenchmark** on the customer's system. Wernham Hogg Labels does not have access to the public MARS apps on the CLOUDFLOW of Future Software, so Rick has to share the MARS app with Wernham Hogg Labels.

## Approach

Rick takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS > My Apps > Installed. This list gives an overview of all the MARS apps that are installed on Rick's local system.
- 3. Next to the MARS app quickbenchmark, he selects Upload to MARS.
- 4. He selects Upload to MARS.
- 5. He opens MARS > My Apps > Available.
- 6. Next to the MARS app quickbenchmark, he selects Details.
- 7. He selects Share with ... and enters Wernham Hogg Labels' customer code: BE-WHL.
- 8. He selects Share.

The MARS app **approvalportalwhl** is now shared on Wernham Hogg Labels' personal space on MARS and can be installed and used by Rick to test their performance. The MARS app cannot be edited.

## Scenario: assigning the co-ownership of a MARS app

These scenarios includes the context and approach for assigning the co-ownership of a MARS app to another user.

## Assigning co-ownership of a MARS app to a customer

This scenario includes the context and approach for assigning the co-ownership a MARS app to a customer.

## Context

**Rick**, a support engineer at **Future Software** has the MARS app **approvalportalwhl** uploaded on his personal space on MARS. The MARS app contains some approval workflows and a customized HTML page for one of his customers, **Wernham Hogg Labels**. In combination with a customized HTML page for Wernham Hogg Labels, the workflows can be used as a portal. Originally, the MARS app was created by **Maggie**, his colleague support engineer at Future Software for one of her own customers, but Rick copied the MARS app to his own system and customized it for Wernham Hogg Labels (see Scenario: copying a public MARS app on page 412.

Rick wants to install the MARS app at the CLOUDFLOW system at Wernham Hogg Labels, so they can use it. However, it is possible that some time after installation, Rick will need to edit the MARS app. Therefore, he chooses not to share the MARS app, but instead to assign the co-ownership to the customer. In that case, he will be able to edit the MARS app and upload the new version if needed.

## Approach

Rick takes the following steps:

- 1. He logs in to CLOUDFLOW with his username and password.
- 2. He opens MARS > My Apps > Available. This list gives an overview of all the CLOUDFLOW Applications that are available on Rick's personal space on MARS and that he can share.
- 3. Next to the MARS app approvalportalwhl, he selects Details.
- 4. He selects Add owner ... and enters Wernham Hogg Labels' customer code: BE-WHL.
- 5. He selects Add owner.

The MARS app **approvalportalwhl** is now visible in the list of Wernham Hogg Labels' MARS apps.

## Assigning co-ownership of a MARS app with a colleague

This scenario includes the context and approach for assigning the co-ownership a MARS app with a colleague.

## Context

**Maggie**, a support engineer at **Future Software**, has created the MARS app **approvalportaldmf**. The MARS app contains some approval workflows for one of her customers, **Dunder Mifflin Prepress**. In combination with a customized HTML page for Dunder Mifflin Prepress, these workflows can be used as a portal. The workflows and the HTML page will be very useful for other CLOUDFLOW users as well, so Maggie wants to make the MARS app publicly available on MARS (see Scenario: publishing a MARS app on page 415).

However, Maggie experiences that some of her workflows run into some errors. She tried to solve them, but she keeps on running into the same problems. Therefore, she asks **Rick**, her colleague support engineer at **Future Software**, to help her. Maggie assigns Rick as a co-owner of the MARS app, so he can:

- Install it.
- Investigate the errors.
- Edit the MARS app if needed.
- Upload a new and corrected version of the MARS app.

## Approach

Maggie has not yet uploaded the MARS app **approvalportaldmf** to MARS, so she needs to upload it first:

- 1. She logs in to CLOUDFLOW with his username and password.
- 2. In MARS > My Apps > Installed, she selects Upload to MARS next to the MARS app approvalportaldmf.
- 3. She selects Upload to MARS.

The MARS app approvalportaldmf is now uploaded to Maggie's personal space on MARS.

Now Maggie needs to assign Rick as a co-owner of the MARS app:

- 1. She opens MARS > My Apps > Available. This list gives an overview of all the CLOUDFLOW Applications that are available on Maggie's personal space on MARS and that she can share.
- 2. Next to the MARS app approvalportaldmf, she selects Details.
- 3. She selects Add Owner ... and enters Rick's customer code: BE-RICK.
- 4. She selects Add Owner.

The MARS app **approvalportaldmf** is on Rick's personal space on MARS.

- 1. Rick opens MARS and has access to the MARS app approvalportaldmf.
- 2. He selects Install.
- **3.** He finds the errors and makes the necessary changes.
- 4. He uploads a new version of the MARS app, where he changes the version number and the changelog file.
- 5. Since Maggie is also the owner of the MAES app, she has access to the updated version on her personal space on MARS.
- 6. She selects Update to install the updated MARS app on her CLOUDFLOW system.

## Scenario: installing and updating a MARS app at a customer

This scenario includes the context and approach for installing and updating a MARS app that you have created for a customer.

#### Context

**Rick**, a support engineer at **Future Software** has assigned the co-ownership of his MARS app **approvalportalwhl** to one of his customers, **Wernham Hogg Labels** (see Assigning co-ownership of a MARS app to a customer on page 417). The MARS app contains some approval workflows and a customized HTML page for Wernham Hogg Labels. In combination with a customized HTML page, the workflows can be used as a portal.

Since Rick has assigned the co-ownership of the MARS app to Wernham Hogg Labels, it is in their list of MARS apps. Consequently, when Rick is at the customer, he can easily install the MARS app.

#### Approach

Rick takes the following steps:

- 1. He logs in to CLOUDFLOW with the credentials of Wernham Hogg Labels.
- 2. He opens MARS > My Apps > Available. This list gives an overview of all the MARS apps of Wernham Hogg Labels that are available on their personal space on MARS.
- 3. Next to the MARS app approvalportalwhl, he selects Install.

The MARS app and all its assets and workflows are now installed on the local CLOUDFLOW system at Wernham Hogg Labels and can be used.

A few weeks later, Rick is back at Wernham Hogg Labels to make some changes in the MARS app and to upload a new version of the MARS app, containing the changes.

He takes the following steps:

- 1. He logs in to CLOUDFLOW with the credentials of Wernham Hogg Labels.
- 2. He opens MARS > My Apps > Installed. This list gives an overview of all the MARS apps of Wernham Hogg Labels that are installed on their local system.
- 3. In MARS > My Apps > Installed, he selects Edit next to the MARS app approvalportalwhl.
- 4. He changes the version number of the MARS app. He fixed some bugs in the approval workflows and the MARS app stays backwards compatible, so he increments the Patch number from 0.0.1 to 0.0.2.
- 5. He selects Save. As a consequence, the version of the MARS app approvalportalwhl in MARS > My Apps > Installed and the version of the same MARS app in MARS > My Apps > Available is different. So it is very important that Rick uploads the MARS app to MARS in order not to lose the bug fixes in the workflows.
- 6. In MARS > My Apps > Installed, he selects Upload to MARS next to the MARS app approvalportalwhl.

approvalportalwhl is now uploaded to Wernham Hogg Labels' personal space on MARS.

# **CLOUDFLOW Plug-in Suite**

With CLOUDFLOW Plug-in Suite you can interact more easily between CLOUDFLOW and files stored on disk.

CLOUDFLOW Plug-in Suite consists of the following parts:

- Plug-ins:
  - Illustrator Plug-in runs as a plug-in in Illustrator and lets you interact between Illustrator and CLOUDFLOW.
  - PACKZ Plug-in runs as a plug-in in PACKZ and lets you interact between PACKZ and CLOUDFLOW.
- **Desktop Service** runs on a workstation but does not require a host. Desktop Service lets you interact between CLOUDFLOW and the operating system. For example, Desktop Service can reveal a file in finder from within CLOUDFLOW.

## Installation

How to install CLOUDFLOW Plug-in Suite.

#### Installing CLOUDFLOW Plug-in Suite

Ű.

- 1. Log in to the Data Center and download the \*.pkg file.
- 2. Double click the \*.pkg file to start the installation and follow the installation steps.

## #

- 1. Log in to the Data Center and download the \*.msi file.
- 2. Double click the \*.msi file to start the installation and follow the installation steps.

## Stopping and uninstalling CLOUDFLOW Plug-in Suite

Stop Plug-in Suite:/Applications/cloudflow plug-in suite//nucleusd --stop --frame

Uninstall Plug-In Suite: /Applications/cloudflow\_plug-in\_suite//nucleusd --uninstall -- frame

Note: For more Plug-in Suite commands, you can use command /Applications/cloudflow\_plugin\_suite//nucleusd --help

## **Illustrator Plug-in**

With the Illustrator Plug-in, Illustrator can directly interact with CLOUDFLOW files.

The Illustrator Plug-in provides multiple windows in Illustrator:

- In menu File > CLOUDFLOW
  - Open From Cloudflow
  - Save A Copy To Cloudflow
  - Save To Cloudflow
  - Submit To Cloudflow

- In menu Window > CLOUDFLOW
  - Cloudflow KIOSK
  - Page Boxes
  - Preferences

The Illustrator Plug-in is available on Mac and Windows. The following versions of Illustrator are supported:

- Illustrator CC 2017
- Illustrator CC 2018
- Illustrator CC 2019
- Illustrator 2020

## Connect to the CLOUDFLOW server

Before you can use the Illustrator Plug-in, you have to make a connection between Illustrator and the CLOUDFLOW server.

To do this in Illustrator, go to **Window** > **CLOUDFLOW** > **Preferences**.

Enter the following configuration options:

- Cloudflow Server URL: the URL of the CLOUDFLOW Server.
- Direct File Access: if you select this checkbox, the file will be accessed directly from within the File store when you open in via File > CLOUDFLOW > Open From Cloudflow. If you leave this checkbox unselected, the file you open with File > CLOUDFLOW > Open From Cloudflow is copied in the Work Folder and will be accessed from here.
- Client Name: the name of the CLOUDFLOW client, for File store mapping or mounting (for example MAC\_CLIENT or WINDOWS\_CLIENT). This is to make sure that the plug-ins have access to the correct files on the correct File store.

## Example

You have configured two Work servers in CLOUDFLOW:

- **PP\_WORK\_SERVER**, which contains the CLOUDFLOW Work server configuration and a mapping to a local File store (**PP\_FILE\_STORE**).
- MAC\_CLIENT, which also contains a file store mapping PP\_FILE\_STORE to the File store on PP\_WORK\_SERVER.

If you enter MAC\_CLIENT in Illustrator > Window > CLOUDFLOW > Preferences > Client Name, the Illustrator plug-in has access to the files on the File store on PP\_WORK\_SERVER

- User Name: the User name of the CLOUDFLOW user.
- **Password**: the password to log in with the CLOUDFLOW users.
- Enable Automatic Login: if you select this checkbox, you will be automatically logged in.

If you select Save, the connection between the Illustrator Plug-in and CLOUDFLOW server is established.

## File menu

In the File menu you have access to several Plug-in Suite options.

Note: If you have selected the checkbox Enable Automatic Login in Window > CLOUDFLOW > Preferences, you will be logged in automatically. If you have not selected the checkbox, a login window appears where you need to enter your password and select Login.

## **Open From Cloudflow**

This function activates a window to a CLOUDFLOW File store if a CLOUDFLOW Server connection has been made in **Window** > **CLOUDFLOW** > **Preferences**.

Options:

- You can display the files in Icon view or List view.
- You can use the search field at the top right of the window to search for a file in the CLOUDFLOW File store.

## Save A Copy To Cloudflow

With this function you can save a file that was not opened from CLOUDFLOW in a File store in CLOUDFLOW.

- 1. Browse to the location in the File store where you want to save the file.
- 2. If desired, edit the name of the file in the field on top of the dialog.
- 3. Select Save.

The file now has a CLOUDFLOW URL and is a CLOUDFLOW asset.

## Save To Cloudflow

If you open a file with the function **Open From Cloudflow** and the checkbox next to **Window** > **CLOUDFLOW** > **Preferences** > **Open From Cloudflow** is not selected, a copy of the file that you open is copied in the **Work Folder** and is actually opened from there. In this case, the function **Save To Cloudflow** allows you to save changes in the original file in the CLOUDFLOW File store as well.

## Submit To Cloudflow

If a connection is established in the CLOUDFLOW Server settings in **Window** > **CLOUDFLOW** > **Preferences**, a file that is currently opened in Illustrator can be sent directly to a CLOUDFLOW workflow via the **Submit To Cloudflow** function.

To submit a file that is currently opened in Illustrator to a workflow, follow these steps:

- 1. Select the Workflow you want to submit to.
- 2. Select the Input.
- 3. If needed, fill in the required form fields.
- 4. The tab Host displays the list of all the files that are currently open in Illustrator. For each file, there is an indication whether the file:
  - Is located inside a File store or is located outside a File Store.
  - Has been saved or has not been saved.
- 5. Select the Transfer Type:
  - **Direct**: if you select this option, you interact directly with files inside a File store and that are known by CLOUDFLOW. This option is available in case the checkbox **Enable File store browsing** is selected in the **Start From Kiosk** node.
  - Upload: if you select this option, files outside a File store and that are unknown by CLOUDFLOW will be uploaded first. By uploading them, the files are put in a folder that you have configured in the Upload Location field in the Start From Kiosk node. This way, the files will get a CLOUDFLOW path and become

known by CLOUDFLOW. This option is available in case the checkbox **Enable upload** is selected in the **Start From Kiosk** node.

#### Example

Jpload	Cloudflow	Host		Upload	Cloudflow	Host	
Transfer Type: O Direct O Upload			Transfer	Type: 🔿 Direct 🤇	Upload		
	Name	In File Store	Sa		Name	In File Store	Saved
	File 3	×	1		File 3	×	×
	File 2.ai	×	•		File 2.ai	×	~
	File 1.ai	~	•		File 1.ai	~	~

- File 1 has been saved in a File store. You can directly select it from the File store to submit it to the workflow.
- File 2 has been saved outside a File store. You need to upload the file before you can submit it to the workflow.
- File 3 has not been saved. You cannot submit it to the workflow.
- 6. Select the file(s) you want to submit to a workflow. You can select files with a checkbox, but you cannot select files that are not saved.
- 7. Select Submit to submit the selected file(s) to the workflow. A *jacket* containing a *workable* is created.

## Window menu

#### **Cloudflow KIOSK**

The **Cloudflow KIOSK** window allows you to directly interact between the files that are currently opened in Illustrator and CLOUDFLOW.

To open the Cloudflow KIOSK window, go to Window > CLOUDFLOW > Cloudflow KIOSK.

## Submit a file that is opened in Illustrator to a workflow

To submit a file that is currently opened in Illustrator to a workflow, follow these steps:

- 1. Open Cloudflow KIOSK.
- 2. Enter your password and log in. If you have selected Enable Automatic Login in the Preferences window, you will be automatically logged in.
- 3. Select Create New....
- 4. Select the Workflow you want to submit to.
- 5. Select the Input.
- 6. If needed, fill in the required form fields.
- 7. Select **Host** to display the list of all the files that are currently open in Illustrator. For each file, there is an indication whether the file:
  - Is located inside a File store or is located outside a File Store.
  - Has been saved or has not been saved.

- 8. Select the Transfer Type:
  - **Direct**: if you select this option, you interact directly with files inside a File store and that are known by CLOUDFLOW. This option is available in case the checkbox **Enable File store browsing** is selected in the **Start From Kiosk** node.
  - Upload: if you select this option, files outside a File store and that are unknown by CLOUDFLOW will be uploaded first. By uploading them, the files are put in a folder that you have configured in the Upload Location field in the Start From Kiosk node. This way, the files will get a CLOUDFLOW path and become known by CLOUDFLOW. This option is available in case the checkbox Enable upload is selected in the Start From Kiosk node.

#### Example

	Nama					
	Name	In File Store	Sa	Name	In File Store	Saved
🗆 Fi	ile 3	×	:	File 3	ж	х
🗆 Fi	ile 2.ai	×		File 2.ai	×	~
🗆 Fi	ile 1.ai	~	•	File 1.ai	~	~

- File 1 has been saved in a File store. You can directly select it from the File store to submit it to the workflow.
- File 2 has been saved outside a File store. You need to upload the file before you can submit it to the workflow.
  File 3 has not been saved. You cannot submit it to the workflow.
- **9.** Select the file(s) you want to submit to a workflow. You can select files with a checkbox, but you cannot select files that are not saved.

10. Select Submit to submit the selected file(s) to the workflow. A *jacket* containing a *workable* is created.

#### Jackets overview

If you open **Window** > **CLOUDFLOW** > **Cloudflow KIOSK**, all the jackets are listed that were created by submitting open Illustrator files to a CLOUDFLOW workflow.

Icon	Status
	Finished jackets
×	Jackets in error
11	Jackets on hold
	Running jackets

If you select a jacket, more details are displayed.

#### Filter jackets

You can use the following filters:

• By a query: if you enter a query in the quick search field, the jackets containing the query are displayed.

• By status: if you select a jacket's status filter in the topbar, the jackets in that status are displayed.

🗙 II 🕨 🎮

- **By Modified/Created**: if you select this filter, only the jackets that are created or modified at a specified moment are displayed.
- **By Workflow**: if you select this filter, only the jackets that were submitted to a specific workflow on are displayed.
- By User:
  - All Users: if you select this filter, all jackets of any user are displayed.
  - Just Me: if you select this filter, only the jackets of the current user are displayed.

## Open a file that is on hold and release the workable

If you submit a file to a workflow that has a **Hold in Kiosk** node, the workable is put on hold and you have to make a decision to release the workable and let the workflow continue.

## Example

This workable was put on hold:

File 2		
PDF	Start Current State Finished	1 test - joepie 1 test - Hold In Kiosk normal no
	ОК	NOT OK
FILES		
🖋 File 2.ai		

In this case, you have to make the decision whether the file is **OK** or **NOT OK**. To do that, you open the file in Illustrator by selecting the link. If the checkbox **Use File store Access** is selected in the **Hold in Kiosk** node, the file is opened directly from the **Upload Location** in the File store that you configured in the **Start From Kiosk** node. If the checkbox is not selected, the file is temporary stored in the **Work Folder** and is opened from there.

After you have opened the file and have edited it if needed, you can release the workable by selecting one of the decision buttons.

## Page Boxes

The Page Boxes window allow you to manage the page boxes in Adobe Illustrator.

To open the Page Boxes window, go to Window > CLOUDFLOW > Page Boxes.

## What are page boxes

There might be up to five different descriptions in a PDF that relates to its size. These are called the page boxes:

• Media Box

The Media Box is the largest page box in a PDF. The other page boxes can be equal to the Media Box but they should not be bigger. The Media Box is used to specify the width and height of the page. This probably equals the actual page size for the average user. For prepress use, this is not the case as pages are defined slightly oversized so that the bleed, the crop marks, information panels and other useful information may be visible as well. This means that PDF documents used in graphic arts usually have a Media Box, which is bigger than the trimmed page size.

• Crop Box

The Crop Box defines the region to which the page contents are to be clipped. Adobe Acrobat uses this size for screen display and printing. For prepress use, the Crop Box is irrelevant.

Bleed Box

The Bleed Box determines the region to which the page contents need to be clipped when output in a production environment. Usually the Bleed Box is 3 to 5 millimeters bigger than the Trim Box. Most prepress systems allow you to define the amount of bleed and ignore the Bleed Box.

• Trim Box

The Trim Box defines the intended dimensions of the finished page. Contrary to the Crop Box, the Trim Box is very important because it defines the actual page size. Most imposition programs and workflows use the Trim Box as the base for positioning pages on a press sheet or labels and packaging on a step and repeat.

Art Box

The Art Box can define a region within a page that is of special interest. It is rarely used by applications.

## Use PDF page boxes in Illustrator with CLOUDFLOW Plug-in suite

You can perform the following actions:

• View or hide the page boxes in the artwork.

Select or deselect the checkbox next to a page box to show or hide the page box individually. Select or deselect the **Show All** checkbox to show or hide all page boxes at once.

• View and change the dimensions of the page boxes.

All the page boxes have a **Width** and **Height** input field. If you enter a value, the concerning page box will be created. If you change a value, the dimensions of the page box are changed. In **Reference point** you can select an anchor that defines in which direction the page box is changed.

• Set the dimensions or position of a page box to another page box.

Select the page box of which you want to change the position or dimension. Next to that page box, select a page box from the **From...** drop-down list to which you want to set the page box.

• Copy the dimensions or position of a page box to another page box.

Select the page box of which you want to copy the dimensions or position. On the right side of that page box, select a page box from the **To...** column to which you want to copy the dimensions or position.

## Preferences

In the **Preferences** window you can make a connection to a CLOUDFLOW server and define the parameters of CLOUDFLOW Plug-in Suite.

To open the Preferences window, go to Window > CLOUDFLOW > Preferences.

It has the following configuration options:

- Cloudflow Server URL: the URL of the CLOUDFLOW Server.
- Direct File Access: if you select this checkbox, the file will be accessed directly from within the File store when you open in via File > CLOUDFLOW > Open From Cloudflow. If you leave this checkbox unselected, the file you open with File > CLOUDFLOW > Open From Cloudflow is copied in the Work Folder and will be accessed from here.
- Client Name: the name of the CLOUDFLOW client, for File store mapping or mounting (for example MAC\_CLIENT or WINDOWS\_CLIENT). This is to make sure that the plug-ins have access to the correct files on the correct File store.

#### Example

You have configured two Work servers in CLOUDFLOW:

- **PP\_WORK\_SERVER**, which contains the CLOUDFLOW Work server configuration and a mapping to a local File store (**PP FILE STORE**).
- MAC\_CLIENT, which also contains a file store mapping PP\_FILE\_STORE to the File store on PP WORK SERVER.

If you enter MAC\_CLIENT in Illustrator > Window > CLOUDFLOW > Preferences > Client Name, the Illustrator plug-in has access to the files on the File store on PP\_WORK\_SERVER

- User Name: the User name of the CLOUDFLOW user.
- **Password**: the password to log in with the CLOUDFLOW users.
- Enable Automatic Login: if you select this checkbox, you will be automatically logged in.
- **Packz Application**: the location of PACKZ on the work station.
- Work Folder: the location where the files are temporary stored when they are in a workflow that is put on hold via a Hold in Kiosk node. The files are downloaded to this work folder and are opened from this folder and you can view and edit them. After the file has been edited and released, the file is removed from the folder.
  - [] Important: In case the checkbox Use File Store Access is selected in the Hold in Kiosk node, the file is not stored in this folder. In that case, the file is opened directly from the Upload Location in the File store that you configured in the Start From Kiosk node.

See Open a file that is on hold and release the workable on page 424 for more information about workables that are put on hold.

# PACKZ Plug-in

With the PACKZ Plug-in, PACKZ can directly interact with CLOUDFLOW files.

## Enable CLOUDFLOW Plug-in Suite in PACKZ

You have to enable CLOUDFLOW Plug-in Suite in PACKZ to use it.

To do so, go to **PACKZ** > **Preferences** > **Servers** > **Cloudflow Server** and select the checkbox next to **Use Plug-in Suite**. To establish a connection between PACKZ and the CLOUDFLOW server, make sure that all the CLOUDFLOW Server settings are entered correctly.

Select Setup to open the Preferences window.

- Cloudflow Server URL: the URL of the CLOUDFLOW Server.
- Direct File Access: if you select this checkbox, the file will be accessed directly from within the File store when you open in via File > CLOUDFLOW > Open From Cloudflow. If you leave this checkbox unselected, the file you open with File > CLOUDFLOW > Open From Cloudflow is copied in the Work Folder and will be accessed from here.

• Client Name: the name of the CLOUDFLOW client, for File store mapping or mounting (for example MAC\_CLIENT or WINDOWS\_CLIENT). This is to make sure that the plug-ins have access to the correct files on the correct File store.

## Example

You have configured two Work servers in CLOUDFLOW:

- **PP\_WORK\_SERVER**, which contains the CLOUDFLOW Work server configuration and a mapping to a local File store (**PP\_FILE\_STORE**).
- MAC\_CLIENT, which also contains a file store mapping PP\_FILE\_STORE to the File store on PP\_WORK\_SERVER.

If you enter MAC\_CLIENT in Illustrator > Window > CLOUDFLOW > Preferences > Client Name, the Illustrator plug-in has access to the files on the File store on PP\_WORK\_SERVER

- User Name: the User name of the CLOUDFLOW user.
- **Password**: the password to log in with the CLOUDFLOW users.
- Enable Automatic Login: if you select this checkbox, you will be automatically logged in.
- Packz Application: the location of PACKZ on the work station.
- Work Folder: the location where the files are temporary stored when they are in a workflow that is put on hold via a Hold in Kiosk node. The files are downloaded to this work folder and are opened from this folder and you can view and edit them. After the file has been edited and released, the file is removed from the folder.
  - **Important:** In case the checkbox Use File Store Access is selected in the Hold in Kiosk node, the file is not stored in this folder. In that case, the file is opened directly from the Upload Location in the File store that you configured in the Start From Kiosk node.

See Open a file that is on hold and release the workable on page 424 for more information about workables that are put on hold.

- Save: select this button to save the settings.
- Go To Login: select this button to go to the CLOUDFLOW log in page.
- Check Cloudflow Server: select this button to sync the file store info and check the status of the CLOUDFLOW server.
- File Store: indicates which File store is known.
- **Path**: indicates the File store path.
- Accessible: indicates whether the File store is accessible or not.

## Use CLOUDFLOW Plug-in Suite in PACKZ

When CLOUDFLOW Plug-in Suite is enabled in PACKZ, you have access to various functionalities.

## Open a file from a mapped File store

If you open a file from a mapped File store, PACKZ shows a CLOUDFLOW icon in the bottom of the document page in PACKZ.

There are various possible ways to can open a file in PACKZ from a mapped File store:

- In CLOUDFLOW: in **ASSETS > Open File with Packz**.
- In PACKZ: in File > Open File From Cloudflow.

## Kiosk

The **Kiosk** window allows you to directly interact between the files that are currently opened in PACKZ and CLOUDFLOW.

To open Kiosk in PACKZ, go to Window > Kiosk.

## Submit a file that is opened in PACKZ to a workflow

To submit a file that is currently opened in PACKZ to a workflow, follow these steps:

- 1. Open Window > Kiosk.
- 2. Enter your password and log in. If you have selected **Enable Automatic Login** in the **Preferences** window, you will be automatically logged in.
- 3. Select Create New....
- 4. Select the Workflow you want to submit to.
- 5. Select the Input.
- 6. If needed, fill in the required form fields.
- 7. Select **Host** to display the list of all the files that are currently open in PACKZ. For each file, there is an indication whether the file:
  - Is located inside a File store or is located outside a File Store.
  - Has been saved or has not been saved.
- 8. Select the Transfer Type:
  - **Direct**: if you select this option, you interact directly with files inside a File store and that are known by CLOUDFLOW. This option is available in case the checkbox **Enable File store browsing** is selected in the **Start From Kiosk** node.
  - Upload: if you select this option, files outside a File store and that are unknown by CLOUDFLOW will be uploaded first. By uploading them, the files are put in a folder that you have configured in the Upload Location field in the Start From Kiosk node. This way, the files will get a CLOUDFLOW path and become known by CLOUDFLOW. This option is available in case the checkbox Enable upload is selected in the Start From Kiosk node.

#### Example

		Transfer Type: • Direct Upload		 i)po: OBildet	opioda	
	Name	In File Store	Sa	Name	In File Store	Saved
	File 3	×		File 3	×	×
D     F	File 2.ai	×	•	File 2.ai	×	~
□ F	File 1.ai	~	•	File 1.ai	~	~

- File 1 has been saved in a File store. You can directly select it from the File store to submit it to the workflow.
- File 2 has been saved outside a File store. You need to upload the file before you can submit it to the workflow.
- File 3 has not been saved. You cannot submit it to the workflow.
- **9.** Select the file(s) you want to submit to a workflow. You can select files with a checkbox, but you cannot select files that are not saved.
- 10. Select Submit to submit the selected file(s) to the workflow. A *jacket* containing a *workable* is created.

#### Jackets overview

If you open **Window** > **Kiosk**, all the jackets are listed that were created by submitting open PACKZ files to a CLOUDFLOW workflow.

Icon	Status
	Finished jackets

Icon	Status
×	Jackets in error
11	Jackets on hold
	Running jackets

If you select a jacket, more details are displayed.

## Filter jackets

You can use the following filters:

- By a query: if you enter a query in the quick search field, the jackets containing the query are displayed.
- By status: if you select a jacket's status filter in the topbar, the jackets in that status are displayed.

×		
~		

- **By Modified/Created**: if you select this filter, only the jackets that are created or modified at a specified moment are displayed.
- **By Workflow**: if you select this filter, only the jackets that were submitted to a specific workflow on are displayed.
- By User:
  - All Users: if you select this filter, all jackets of any user are displayed.
  - Just Me: if you select this filter, only the jackets of the current user are displayed.

## Open a file that is on hold and release the workable

If you submit a file to a workflow that has a **Hold in Kiosk** node, the workable is put on hold and you have to make a decision to release the workable and let the workflow continue.

## Example

This workable was put on hold:

•	File 2		
	File 2		
	PDF	Start Current State Finished	1 test - joepie 1 test - Hold In Kiosk normal no
		ОК	NOT OK
	FILES		

In this case, you have to make the decision whether the file is **OK** or **NOT OK**. To do that, you open the file in PACKZ by selecting the link. If the checkbox **Use File store Access** is selected in the **Hold in Kiosk** node, the file is opened directly from the **Upload Location** in the File store that you configured in the **Start From Kiosk** node. If the checkbox is not selected, the file is temporary stored in the **Work Folder** and is opened from there.

After you have opened the file and have edited it if needed, you can release the workable by selecting one of the decision buttons.

# **Desktop Service**

With Desktop Service you can interact between CLOUDFLOW and the operating system of your local machine.

## Configuration

How to configure CLOUDFLOW Plug-in Suite.

## Configuration

You can open the link to configure CLOUDFLOW Plug-in Suite by clicking in the right area of the CLOUDFLOW topbar.

HELLO <b><fullname></fullname></b>
PREFERENCES
MANUAL
API REFERENCE
INSPECT COLLECTIONS
PLUG-IN SUITE SETTINGS
LOGOUT

Select PLUG-IN SUITE SETTINGS to open the settings.

- Cloudflow Server URL: the URL of the CLOUDFLOW Server.
- Direct File Access: if you select this checkbox, the file will be accessed directly from within the File store when you open in via File > CLOUDFLOW > Open From Cloudflow. If you leave this checkbox unselected, the file you open with File > CLOUDFLOW > Open From Cloudflow is copied in the Work Folder and will be accessed from here.
- Client Name: the name of the CLOUDFLOW client, for File store mapping or mounting (for example MAC\_CLIENT or WINDOWS\_CLIENT). This is to make sure that the plug-ins have access to the correct files on the correct File store.

## Example

You have configured two Work servers in CLOUDFLOW:

- **PP\_WORK\_SERVER**, which contains the CLOUDFLOW Work server configuration and a mapping to a local File store (**PP\_FILE\_STORE**).
- MAC\_CLIENT, which also contains a file store mapping PP\_FILE\_STORE to the File store on PP\_WORK\_SERVER.

If you enter MAC\_CLIENT in Illustrator > Window > CLOUDFLOW > Preferences > Client Name, the Illustrator plug-in has access to the files on the File store on PP WORK SERVER

- User Name: the User name of the CLOUDFLOW user.
- **Password**: the password to log in with the CLOUDFLOW users.
- Enable Automatic Login: if you select this checkbox, you will be automatically logged in.
- **Packz Application**: the location of PACKZ on the work station.
- Work Folder: the location where the files are temporary stored when they are in a workflow that is put on hold via a Hold in Kiosk node. The files are downloaded to this work folder and are opened from this folder and you can view and edit them. After the file has been edited and released, the file is removed from the folder.
  - **Important:** In case the checkbox Use File Store Access is selected in the Hold in Kiosk node, the file is not stored in this folder. In that case, the file is opened directly from the Upload Location in the File store that you configured in the Start From Kiosk node.

See Open a file that is on hold and release the workable on page 424 for more information about workables that are put on hold.

- Save: select this button to save the settings.
- Go To Login: select this button to go to the CLOUDFLOW log in page.
- Check Cloudflow Server: select this button to sync the file store info and check the status of the CLOUDFLOW server.
- File Store: indicates which File store is known.

- Path: indicates the File store path.
- Accessible: indicates whether the File store is accessible or not.

## **Functionalities**

These are the functionalities of the Desktop Service of CLOUDFLOW Plug-in Suite:

- ASSETS > Open Folder in OS
- ASSETS > Reveal File in OS
- ASSETS > Open File with Packz
- PAGEBUILDER > Actions > Open Folder
- PAGEBUILDER > Actions > Reveal File

In the CLOUDFLOW Top bar, you can select the link to the **CLOUDFLOW PLUG-IN SUITE SETTINGS**, where you can define the preferences for the CLOUDFLOW Plug-in Suite settings. See CLOUDFLOW Plug-in Suite **SETTINGS** for more information.

# COCKPIT

With COCKPIT you can manage the labels and packaging process from the order stage to the production stage.

It is a MARS App that contains a set of templates to build job driven solutions for labels and packaging. A connection to ERP/MIS solutions will improve the work efficiency.

#### Goal of COCKPIT in a labels or packaging process

At the start of a usual label or packaging process, two things happen:

- The sales people place an order, which is stored as an item in an ERP or MIS system
- The customers provide you with files, which are stored in a file system.

COCKPIT allows you to connect the files with the associated items. After the association between the file and the items, you can start the regular workflow in order to prepare the file for production.

#### Content

Print Production Cockpit includes:

- Customer File Management.
- Item Management with customer approval.
- Job Management for label and folding carton automated step and repeat.

Print Production Cockpit is a combination of the following CLOUDFLOW modules:

- WORKSPACE
- PAGEBUILDER
- JOBS
- PROOFSCOPE
- DATALINK
- PACKZflow

**Note:** You need a license for all these modules to be able to use Print Production Cockpit. Contact your local sales representative for more information.
# Install COCKPIT

This chapter explains how to install COCKPIT.

COCKPIT is a Public MARS app. See MARS on page 398 for more information on MARS and MARS apps.

To install COCKPIT, follow these steps:

- 1. Go to MARS > Public.
- 2. Go to the MARS app COCKPIT and select Install.

After installation, the MARS app COCKPIT should be in your list of Installed MARS apps.

# **Initiate COCKPIT**

Before the first use of COCKPIT, you need to initialize it.

- 1. Open KIOSK.
- 2. Select Create New....
- 3. Select the workflow ppc-main. You can also do this in the COCKPIT ADMIN.
- 4. Fill in the content of the context field: ppc.
- 5. Select Submit.

# Use COCKPIT

To open the Welcome page of COCKPIT, follow these steps:

- 1. Go to ASSETS > PP\_FILE\_STORE > ppc.
- 2. Select start.html.
- 3. Select View File.

COCKPIT consists of the following tabs:

- **HOME**: the Home page.
- UPLOAD: in this tab you can upload the files.
- DELIVERIES: in this tab you have an overview of the files that were uploaded in DELIVERIES.
- **ITEMS**: in this tab you have an overview of the items in your MIS or ERP system.
- JOBS: in this tab you have an overview of the jobs in your MIS or ERP system.
- ADMIN: in this tab you can set the context of COCKPIT and view the logs.

## UPLOAD

In UPLOAD you can upload the files you receive in order to register them.

To upload a file, follow these steps:

- 1. Open the tab UPLOAD.
- **2.** Fill in the fields:
  - Company: the company that has sent you the file you want to upload. This field is required.
  - E-Mail: the email address of the sender of the file. This field is required.
  - Item Number: the original item number of the file, used as a reference by the customer. Mostly, this is a part of the file name or file description.
  - Item Description: the item description.
  - **Project Description**: the project description.
  - **Remarks**: additional remarks.
- **3.** Browse to the file(s) you want to upload.

#### 4. Select Submit.

The file(s) you have uploaded are now registered and are known in the tab **DELIVERIES**.

For each delivery, a subfolder in the **Deliveries** subfolder in ppc\_data is created, where the corresponding delivered file is stored.

## DELIVERIES

DELIVERIES shows a list of the deliveries and the uploaded files.

There are two tabs:

- ACTIVE: this tab shows the deliveries that contain files that are not linked to an ITEM.
- **ARCHIVED**: this tab shows the deliveries of which all the files are **linked** to an **ITEM** and consequently will be used in prepress.

## Create a manual delivery

If you select **New delivery**, you create a manual delivery. This action creates an empty subfolder in the **Deliveries** subfolder in ppc\_data, in which you can manually copy one or multiple files, without uploading them in **UPLOAD**. Make sure to select **Scan Folder** in the delivery details, so the files are added.

## Columns

The following columns are shown:

- **Preview**: the preview icon of the delivered file(s). If multiple files were uploaded, one preview is displayed.
- Delivery id: the id of the delivered file(s). This id is created by COCKPIT when uploading the file(s).
- **Delivery Date**: the date when the file(s) were delivered.
- **Customer item id**: the customer reference id of the delivered file(s). You can enter this id when uploading the file(s) in **UPLOAD**.
- **Customer name**: the name of the customer who has delivered the file(s). You can enter the customer name when uploading the file(s) in **UPLOAD**.
- **Managed by**: the email address of the person who manages the delivery. You can enter the this information when uploading the file(s) in UPLOAD.
- Remarks: the remarks that you have added when uploading the file(s) in UPLOAD.
- Status: the current status of the delivery.

## Sorting and filtering

- Select the column header to sort on the content of the column.
- Enter content in the field underneath the column header to narrow down the data in the column.

## Details of the delivery

If you select the disclosure button, the details of the delivery and the uploaded file(s) are displayed. Per file, the following columns are shown:

- **Preview**: the preview icon of the delivered file(s).
- Tag: the file tag. If you label a file as a **Production** file, a **Production** tag is added.
- **Preflight**: the preflight result of the file. If you label a file as a **Production** file, the file is preflighted and the result is shown in this column.
- Linked to: the item where the file is linked to.

## Actions

If you select the checkbox next to a file, you can select the following options to perform an action:

- Link: if you select this action, a dialog pops up where you can select an item to link the file to.
  - 1. Select the item where you want to link the file to.
  - 2. select Assign to link the file to the selected item.

When the file is linked to an item:

- The file **Status** is updated to **Linked**.
- The delivery is moved to the list of **ARCHIVED** deliveries.
- An item subfolder Original in the Items subfolder in PP\_FILE\_STORE is created, to which the file is copied.
- Reject: if you select this action, the file is rejected and is moved to the list of ARCHIVED deliveries.
- Assign the **Production** tag: if you select this tag, the selected file is preflighted. After preflight, the file is tagged as a **Production** file, and the preflight result is shown in the column **Preflight**.
- Assign the Annexes tag: if you select this tag, the file is tagged as not to be used for production.
- Scan Folder: if you select this action, the deliveries folder is scanned for new files. This is important when you have created a manual delivery.
- Open Folder: if you select this action, the Deliveries subfolder opens.

If you double click a row of a delivered file, a window is opened in which you can:

- Specify additional details for the delivered file or edit the existing details.
- Specify a To Do list.
- View the history of the file.

If you select the preview of a delivered file, the file opens in PROOFSCOPE. A preflight report is available that indicates the warnings, info and errors found in the delivered file.

## ITEMS

**ITEMS** shows a list of the items in your MIS or ERP system.

DATALINK links COCKPIT to an MIS or ERP system that contains all the registered items. Consequently, the content of the **ITEMS** page is coming from the items in the linked MIS/ERP system.

- Select the column header to sort on the content of the column.
- Enter content in the field underneath the column header to narrow down the data in the column.

#### Link a file to an item

If a file in **DELIVERIES** has not been linked to an item yet, the item has status **Wait for Customer Files**. To link a file to an item:

- 1. Go to DELIVERIES.
- 2. Select one of the delivered PDF files.
- 3. Select Link to open the dialog where you can select the item to link the file to.
- 4. Select the item.
- 5. Select Assign.

In **ITEMS**, the item has a preview and has changed to **Waiting for Prepress**. When you open the disclosure triangle of the item, the linked production file is displayed.

#### Flow

This is the flow in **ITEMS**:

- 1. If the item does not have any linked files, the Item Status is Wait for Customer Files.
- 2. If one or multiple files are linked to an item, the status changes to Waiting for Prepress. The Action Edit file is available.

- **3.** Start editing the file in PACKZ:
  - **a.** Select **Edit file**. The status changes to **Edit in progress** and a folder **Production** is created in the Items subfolder, on the same level as the **Original** subfolder. This will allow you to compare the original file with the edited production file.
  - **b.** Go to **PACKZ** > **Window** > **Kiosk**.
  - **c.** Select the jacket. Since the workable is on hold, you can use the filter **Hold** (**III**) to narrow down the jackets list result.
  - d. Open the file by selecting the link in the section FILES.
  - e. Make the edits.
  - f. Select Item Ready.
  - g. Select Save And Submit. The workable continues.
- **4.** Go back to the item in the ITEMS list in COCKPIT. Under the disclosure button both the original file and the edited production file are available.
- 5. The item status has changed to Send to QA. Two Actions are available: QA approve and QA reject.
- 6. Before you can approve or reject the file, select the link **Diff**. This opens the file opens in PROOFSCOPE, where you can compare the edited production file with the original file.
- 7. Go back to the ITEMS list in COCKPIT and select an Action: QA approve or QA reject.
  - If you select **QA approve**, the file is QA approved and the customer approval starts.
  - If you select **QA reject**, the file is QA rejected. The status changes to **Correction** and the file needs to be corrected. After correcting the file, select Action **Send to QA** to resend the file for QA approval.
- **8.** If the file is internally approved with the Action **QA approve**, the customer approval starts. The customer receives an email with a link to open the file in PROOFSCOPE and approve or reject the file with the decision buttons.
  - If the customer approves the file, the status changes to **Ready for Production**.
  - If the customer rejects the file, the status changes to **Correction** and the file needs to be corrected. After correcting the file, select Action **Send to QA** to resend the file for QA approval.

## JOBS

**JOBS** shows a list of the jobs in your MIS or ERP system.

DATALINK links COCKPIT to an MIS or ERP system that contains all the registered items. Consequently, the content of the **JOBS** page is coming from the jobs in the linked MIS/ERP system.

- Select the column header to sort on the content of the column.
- Enter content in the field underneath the column header to narrow down the data in the column.

As soon as all the files in a job are ready for production, the status is **Waiting for Form** and the Step and Repeat can be created. To do so:

- 1. Select the action **Create Form**. An extra subfolder **Forms** is created in the Items subfolder, containing the XML with the repetition information.
  - Double click in the job row for information about the repetition, paper size... This information comes from the MIS/ERP system and is read from the XML file.
  - Open the disclose button to view the repetitions. Click in a repetition row to open the repetition in PROOFSCOPE.
- 2. The status has changed to Waiting for Approval, and two Actions are available: Approve Reject.
  - If the repetition is approved, it is ready for production. The status has changed to **Ready for Production**.
  - If the repetition is rejected, the repetition needs to be corrected. The status has changed to **Correction**. The repetition needs to be recreated with the action **Create Form**.

## TOOLS

TOOLS shows a list of the tools in your MIS or ERP system, such as diecuts, cutter, braille maker...

## **ADMIN**

In ADMIN you can set the context of COCKPIT and view the logs.

## SETUP

In this tab you can set the context of COCKPIT:

- 1. Fill in the content of the context field: ppc.
- 2. Select Submit.

## LOG

In this tab you can view the logs in COCKPIT.

You can sort and filter on all available columns.

- Select the column header to sort on the content of the column.
- Enter content in the field underneath the column header to narrow down the data in the column.

# **Template and customizations**

You can use COCKPIT as a template and customize it.

COCKPIT is available as a public App in MARS. You can use it in two ways:

## As is

You can use COCKPIT as is. This means that you download and install COCKPIT from MARS and do not change it. Consequently, the link with the original MARS App remains and you will be able to update your installed COCKPIT version when a newer version is available on MARS.

## As a template

You can use COCKPIT as a template and customize it according your or your customers needs. This means that you copy COCKPIT to your private space on MARS and customize it according to your or your customer's needs.

In the case of COCKPIT, a part of the folders of the MARS App are copied and a part of the folders are not copied and use the generic resources that are provided through the MARS App.

- The resources in the folders that are copied are no longer linked to the original MARS App and cannot be updated.
- The generic resources in the folders that are not copied can be updated when there is a newer version available on MARS.

Please refer to the documentation in the COCKPIT App on MARS how to use COCKPIT as a template.

# TECTONICS

TECTONICS is a tool designed for viewing, inspecting, cropping and stitching one-bit TIFF or LEN files.

# **Prerequisites**

TECTONICS is a CLOUDFLOW core module.

## **Cloudflow Requirements**

Workspace

• TECTONICS worker to be added to the work server.

# Configuration

#### 1. Check License

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click Settings > License.
- 3. Check whether the Tectonics license is listed.
- 4. If not, click the Update License button.
- 5. If the license is still not listed, request it from your CLOUDFLOW Support contact.

#### 2. Check Worker

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click Settings > Workers.
- 3. Check whether the **Tectonics** worker is listed.
- 4. If not, click the Add Worker button and in the Type drop-down list, select this worker.
- 5. Switch on the Active Worker control and click Create.

# **User interface**

TECTONICS consists of multiple tab sheets.

## Placement

In Placement you can create TECTONICS jobs and fill sheets.

It consists of two sections:

- JOBS: in this section you can create TECTONICS jobs and place the LEN or TIFF files on the sheet.
- SHEETS: in this section the sheets with placed LEN or TIFF files are shown.

#### JOBS

In the JOBS part you can create a TECTONICS job and place the LEN or TIFF files on the sheet.

#### Overview of the jobs

The jobs overview shows the jobs list in the selected queue.

**Note:** The breadcrumb on the top of the page indicates which queue is selected. To select another queue, select **Set Queue** and select one of the available queues by double clicking it. Consequently, the jobs overview of the newly selected queue is displayed.

The jobs list shows multiple columns:

Jobs	5				
🕂 Cr	reate Job 🚦 Place 🛍	j Delete			
					E
0			All	✓ All	✓ All possibil ✓
□・	YoYo_Tectonics _p1	ACE114D	1140 µm	2540 dpi	Placed
	Tectonics_test 0/1	ACE114D	1140 µm	2540 dpi	Incomplete
	A	B			C

- A: A column with the job name
- **B**: The plate designation. This is a unique code that refers to a specific plate type.
- C: The job status.

To filter the jobs list, you can enter a search string in one of the header text fields on top of the columns (D) or use the drop-down list to filter on the job status (E).

Select the disclosure icon to open the details and the TIFF or LEN files of the job. The following information is displayed:

- The plate and job details.
- Composite Display. If you select this option, the composite job will be displayed in PROOFSCOPE.
- The list of LEN or TIFF files of the job. Select a LEN or TIFF file to open it in PROOFSCOPE.

## Create a job

To create a TECTONICS job, follow these steps:

- 1. Select + Create Job.
- **2.** Enter or select the parameters:
  - **a.** Name: here you can define a name for the job.
  - **b.** Queue name: here you can select the Queue for the job. Select the search icon to open the queues list. Double click to select a queue.
  - c. Rotation: here you can select the rotation degree of the LEN or TIFF files. TBA
- **3.** Select **Upload** to upload LEN or TIFF files from outside a CLOUDFLOW File store. Select **Cloudflow** to select LEN or TIFF files from a File store.
- 4. Select Submit.

## Place the LEN or TIFF files

To place the LEN or TIFF files of a job to a sheet, follow these steps:

- 1. Select the checkbox next to a job.
- 2. Select Place.

- 3. A dialog window appears.
  - If you select the option Add to sheets, create new if needed, the LEN or TIFF files will be added to an exiting sheet if they fit, of they will be placed on a new sheet in case they don't fit.
    - Sheet Name: here you can define the sheet name.
    - **Plate**: here you can select the plate type. Select the search icon to open the plates list. Double click to select a plate.
  - if you select the option Add to existing sheet, the LEN or TIFF files will be added to an existing sheet.
    - Sheet Name: here you can select the sheet to add the files to. Select the search icon to open the sheets list. Double click to select a sheet.
- 4. Select Submit.
- 5. The status of the job will change from Incomplete to Placing and finally to Placed.

#### Archive the job

If you select the checkbox next to a job and select Archive, the job is archived.

#### SHEETS

After the LEN or TIFF files of a job are placed on a new or an existing sheet, the sheet appears in the sections SHEETS.

Double click a sheet to open it in PROOFSCOPE.

The following information is displayed around the sheet:

- The dimensions.
- The fill percentage. The minimum fill percentage for the queue can be defined in the queue settings in Queues on page 440. If the minimum fill percentage has not been reached, the percentage will be displayed in orange.
- • : this icon is displayed when the cut mark file is generated. You can double click it to open the cut mark file in PROOFSCOPE.
- Im: this icon is displayed when the TIFF file is generated. You can double click it to open the TIFF file in PROOFSCOPE.
- *•*: this icon is displayed when the sheet has been sent to an output device.

Select the sheet to activate it. You can perform the following actions:

- Unplace: select this option to unplace the LEN of TIFF files of a job from the sheet.
- **Make Partial**: select this option to make a partial plate and cut off the part that is not filled from the plate. This cutoff is added to the database. In the tab Cutoffs on page 440 there is an overview of all the cutoffs.
- Use Cutoff: select this option to search the database for a cutoff that can fit the content of the sheet in case the sheet is not completely filled. The system will automatically use the cutoff with the most optimal fill percentage.
- Send: select this option to send the sheet to an output device.
- Archive: select this option to archive the sheet. This is only possible if the sheet has already been sent to an output device.

## Queues

Queues gives you an overview of the queues and their details.

To filter the queues list, you can enter a search string in one of the header text fields on top of the columns or use the drop-down list to filter on the queues that are enabled or disabled.

## Cutoffs

Cutoffs gives you an overview of the cutoffs.

Cutoffs can be added in two ways:

- Automatically. This happens when a sheet is not completely filled and the option Make Partial was selected. In case a plate is made a partial plate, the part that is not filled is cut off from the plate and added to the database.
- Manually. To do this, follow these steps:
  - 1. Selecting the option Add Cutoff.
  - **2.** Enter the parameters:
    - Cutoff Identifier: here you can define a unique ID for the cutoff.
    - **Plate Designation**: here you can select the plate designation. Select the search icon to open the designations list. Double click to select a designation.
    - **Cutoff Width**: here you can define the cutoff width.
    - Cutoff Height: here you can define the cutoff height.
    - Expiration: here you can define after how many days the cutoff TBA
    - RemarksTBA
  - 3. Select Add.

If you select the checkbox next to a cutoff, you can edit or delete it.

## Archive

Archive gives you an overview of the archive.

## **Statistics**

Statistics allows you to display statistics.

Select an option in the drop-down lists and select Show Statistics to display a page with the statistics.

## Settings

In settings you can define the TECTONICS settings.

It contains multiple sections.

## **Queues Setting**

In this section you can manage queues. A queue can be defined as a combination of a specific plate and an output device.

To add a queue, follow these steps:

- 1. Select Add Queue.
- **2.** Enter the parameters:
  - Name: here you can define the name of the queue.
  - **Plate Designation**: here you can select the plate designation. A plate designation is a unique ID that identifies a plate. Select the search icon to open the designation list. Double click to select a plate designation.
  - **Output Device**: here you can define the output device.
  - Minimum plate fill percentage: here you can define the minimum fill percentage of the sheet.
  - **Timeout (in hours)**: here you can define a timeout. When the timeout is reached, the following happens:
    - In case there is an incomplete job, which means that the job contains LEN or TIFF files that have not been placed on a sheet yet, the TIFF or LEN files are placed automatically on a sheet, and the sheet is sent automatically to the output device.
    - In case there is a sheet that has not been sent to an output device, the sheet is sent automatically to the device.
  - Top Margin: here you can define the top margin of the sheet.
  - Bottom Margin: here you can define the bottom margin of the sheet.
  - Left Margin: here you can define the left margin of the sheet.
  - Right Margin: here you can define the right margin of the sheet.

#### 3. Select Add.

To enable, disable or delete a queue, select the checkbox next to a queue and select **Enable Queue**, **Disable Queue** or **Delete Queue**.

To filter the queues list, you can enter a search string in one of the header text fields on top of the columns or use the drop-down list to filter on the queues that are enabled or disabled.

#### **Plates Settings**

In this section you can manage plates. Select the disclosure button to show the plate details.

To add a plate, follow these steps:

- 1. Select Add Plate.
- **2.** Enter the parameters:
  - Manufacturer: here you can define the plate manufacturer.
  - Name: here you can define the plate name.
  - Thickness: here you can define the plate thickness.
  - Designation: here you can define the plate designation. This is a unique ID that identifies a specific plate.
  - Typical Applications: here you can select an application field.
  - Width: here you can define the plate width.
  - **Height**: here you can define the plate height.
- 3. Select Add.

To delete a plate, select the checkbox next to a plate and select **Delete Plate**. To add or delete a plate size to or from an existing plate, select the checkbox next to a plate and select **Add Plate Size** or **Delete Plate Size**.

To filter the plates list, you can enter a search string in one of the header text fields on top of the columns or use the drop-down list to filter on application field.

#### Preferences

In this section you can manage the TECTONICS settings.

- JOBS AND SHEETS
  - JOBS Path: here you can define the output path of the jobs.
  - Patches fill orientation: here you can define the fill orientation of the LEN or TIFF files.
  - Sheets Path: here you can define the output path of the sheets.
- OUTPUT INFORMATION
  - TIFF Output Path: here you can define the output path of the TIFF files.
  - **PDF Overview Output Path**: here you can define the path for the PDFOverview file.
- CUTTING INFORMATION
  - **Cutting Output Type**: here you can select the cutting output type. You can select manual, ZUND or Kongsberg cutting.
  - Cutting Output Path: here you can define the cutting output path.

#### MOUNTING INFORMATION

- Mounting Method: here you can define the default mount method. Options:
  - Automatic mounting (MOM file): select this option in case the patches will be mounted automatically on the carrier in the mounting machine. A MOM XML file that contains the coordinates of the patches will be exported .
  - **Mirror mounting**: select this option in case the mounting of the patches will be done by mirror mount. A mirror proof PDF will be exported. This file is printed and taped onto the proofing drum of the mounting machine. It will display through a mirror and the marks allow for the patched job to be positioned exactly.
  - Mirror and Automatic (MOM file) mounting: select this option if you want to combine the two options above.
  - **Drill mounting**: select this option in case the mounting of the patches will be done by drill mount.
  - CSV mounting: select this option in case the mounting of the patches will be done based on a CSV file.
  - **DRL mounting**: select this option in case the mounting of the patches will be done based on a DRL file.
- MOM Output Path: here you can define the output path for MOM XML output file.
- Mirror Proofs Output: here you can define the output path for the mirror proof PDF.
- Drill Mount Output Path: here you can specify the output path for the drill mount.
- INTERNATIONAL
  - Length Unit: here you can define the length unit.
  - Small Length Unit: here you can define the unit for small lengths.
  - Resolution Unit: here you can define the resolution unit.
- RESET
  - **Reset Tectonics**: here you can select to reset TECTNONICS, which means that all jobs, sheets and generated files will be removed. This function is used for demo purpose only.

# **INTELLIGENT FLEXO**

# Description

#### Flexography

Flexography or flexo printing is a printing method that uses rubber or flexible relief plates and fast-drying inks to print on a variety of substrates. Owing to flexo's high flexibility the variety of suitable media is so wide that flexo printing is a popular choice in the packaging industry.

But Flexo is not without its challenges, among which are:

- Maintain seamless degradation to 0%.
- Better ink layer on difficult substrates.
- Get control of ink transfer. Locally and all over. Reduce and gain ink transfer.
- Avoid edge void on barcodes and horizontal lines.
- Avoid flexo edges and keep small text sharp.
- Improve typical low opacity of white in Flexo.
- Avoid spread when high volume aniloxes are used.
- Ink cost

Intelligent Flexo was designed specifically to tackle such challenges.

#### **Intelligent Flexo's Benefits**

INTELLIGENT FLEXO achieves its results from an innovative concept, based on **intelligently** identifying problem areas, and applying techniques to improve print quality. These techniques are totally agnostic in that they run independent from their environment, from any plates, any vendor, or any press.

# **Intelligent Flexo Process**

INTELLIGENT FLEXO applies smart algorithms that enhance screening for flexo printing.

#### Screening

INTELLIGENT FLEXO processes one or more incoming TIFF files and outputs these to a dedicated folder. The node applies algorithms that will process the images in a unique manner. First, it intelligently identifies different zones in the artwork that warrant intervention and then processes them according to their respective needs. The intention of these effects is to help overcome (or mask) issues that are inherent to the flexo process.

#### Zones

The zones it enhances are :

- flat paints
- rasters
- barcodes
- text
- borders
- non-printing patterns can be applied to specific regions of a TIFF to permanently mark customer specific plates in a non-erasable, non-printable manner (e.g. to protect mounting mark areas).

All zones are re-screened for all their separations.

## **Prerequisites**

The system requires tuning of the customer's Flexo process + the INTELLIGENT FLEXO software module on CLOUDFLOW.

#### 1. Cloudflow Requirements

Workspace

a

· Intelligent Flexo workers added to the work server. 2 workers per license allowed

#### 2. Setup Requirements

- Consultancy by a technical expert
- Calibration, evaluation, feedback.
  - Attention: INTELLIGENT FLEXO is more of a project than a simple node. Its algorithms involve many parameters and settings and these relate to all of a customer's materials (plates, inks, presses, washing routines,...) and job specs. To fully enjoy its benefits these need first to be meticulously fine-tuned.

#### 3. Screening Requirements

- File format: 1-Bit screen TIFF
  - **Tip:** LEN files can be converted to TIFF using the **Len To TIFF** node that can be found in **CLOUDFLOW** > **Flows** > **Nodes**.

# Configuration

#### 1. Check License

Licenses should be made available by going to the Settings > License page and clicking Update License.

#### 2. Check Worker

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click Settings > Workers.
- 3. Check whether the Intelligent Flexo worker is listed.
- 4. If not, click the Add Worker button and in the Type drop-down list, select this worker.
- 5. Switch on the Active Worker control and click Create.

# **Intelligent Flexo Node**

With the node INTELLIGENT FLEXO you can apply screening and effects that are highly customised to your production requirements to guarantee optimal plate making and printing quality.

Intelligent Flexo		(?)
TIFF files: Processed files:	All files from node previous Node Folder Path Filename w/o extIF Filename ext. Overwrite existing file	0
Patterning		
	Fill with pattern	
Cell wall:	0 µm	
Skip dots smaller than:	0 µm	
Pattern image:		Ø
	Close Sa	ve

The node processes one or more incoming TIFF files and outputs these to a dedicated folder. These settings can be freely configured with the Expression Builder.

i

Tip: Compatibility

As the INTELLIGENT FLEXO node optimises TIFF files, its functionality can be used in conjunction with the Len To TIFF and Merge TIFF files nodes.

It has the following configuration options:

#### Input/Output Info

TIFF files: This field displays the input TIFF files that will be screened. Select the pencil to open Expression Builder.

**Processed files**: This field displays the output TIFF files that will be screened and their location. Select the pencil to open Expression Builder.

Overwrite existing file: Select this check box to overwrite previously screened files.

If this box is not checked and a new TIFF file is generated that has the same name as a previous one, the node will throw an error.

#### Patterning

A TIFF's dots can be printed either 'AS IS' or with additional screening to enhance output quality. In this section you define the surface patterning that you want to apply to the TIFF files. Non-printing patterns can be applied to specific regions of a TIFF to permanently mark customer specific plates in a non-erasable, non-printable manner (e.g. to protect mounting mark areas).

Fill with pattern: Select this check box to enable pattern screening.

**Cell wall**: This field defines the cell wall width. Giving a dot a solid cell wall prevents excess ink from spilling out and the dots from blurring.

**Skip dots smaller than**: This field defines which dots are to be ignored for pattern screening on the TIFF files. Dots that are too small are unsuitable for practical use in that they will fail to be properly exposed onto the plate, or fail to hold and transfer ink properly, or get damaged during the washing of the plate.

**Pattern image**: This field displays the file path to the 1-Bit TIFF image that you want to use for patterned screening. Select the pencil to open Expression Builder.



#### Note: Patterns

Patterns used can be common, specific designs already in use, or newly customized images combining the benefits of your experience and the power of HYBRID's advanced technology.

# **Building a Workflow**

A workflow is a sequence of predefined steps, which are configured with a specific set of parameters, and are executed step after step. They are built with **Nodes**.

#### 1. Starting a Workflow

To build a workflow, proceed as follows:

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click FLOWS.
- 3.
- In the top bar, click Add workflow
- 4. Select the Nodes tab.
- 5. Select a node and drag it onto the workflow area.
- **Note:** Full information on building an editing work flows can be found in the CLOUDFLOW User Manual, in the chapter Workflows on page 117 and the topic Building a workflow on page 117.

#### 2. Demo Workflow

To demonstrate how INTELLIGENT FLEXO operates, suffice it to build the following demo workflow:

- 1. Start a workflow, following the procedure above 1. Starting a Workflow on page 446.
- 2. In the Input section, select the Start from Kiosk node and drag it onto the workflow area.
- 3. In the RIP section, select the Intelligent Flexo node and drag it onto the workflow area.
- 4. In the General Workflow Constructs section, select the End node and drag it onto the workflow area.
- **5.** Connect the nodes as described in the CLOUDFLOW User Manual, in the topic Building a workflow on page 117.

This demo workflow will allow you to experiment with the performance of INTELLIGENT FLEXO. For more elaborate testing facilities, please use the MARS app Intelligent Flexo Test Kit.

# VDP

VDP or **Variable Data Printing** is a tool designed to apply varying layout data from one printed piece to another during one print run without stopping.

**VDP (Variable Data Printing)** is a type of digital printing where data of the layout (such as text, images, colors, barcodes) may change from one printed piece to another during one print run without stopping. The variable data is extracted from an external file.

VDP creates different pages and saves the result as an optimized single or multipage PDF file.

VDP is run from the "VDP" button in the left sidebar.

# Prerequisites

VDP is a CLOUDFLOW module.

## **Cloudflow Requirements**

- Workspace
- VDP engine worker to be added to the work server.

# Configuration

## 1. Check License

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click **Settings** > **License**.
- 3. Check whether the VDP Execute license is listed. This entitles you to up to 8 workers.
  - An optional VDP Booster license entitles you to an extra 8 workers per license.
- 4. If not, click the Update License button.
- 5. If the license is still not listed, request it from your CLOUDFLOW Support contact.

## 2. Check Worker

- 1. Run CLOUDFLOW.
- 2. In the left sidebar menu, click Settings > Workers.
- 3. Check whether the VDP engine worker is listed.
- 4. If not, click the Add Worker button and in the Type drop-down list, select this worker.
- 5. Switch on the Active Worker control and click Create.

# **User interface**

VDP consists of one tab sheet.

<ul> <li>→ Home</li> <li>↓ Notifications</li> <li>↓ Accests</li> </ul>	Processing Jobs Archived Jobs Ac	tivity Log				П	Queue is RUN	IING	
Approval	C ♥ ✓ ∧ ♠ X Abort	Database			State Description		Start Item	Number of	Maximum of Items
Tasks Jobs								iterns -	
RIP		Counter		Showing	tropared 1 to 1 of 1 rows		1	0	
Flows	Processed Jobs								
VDP Tectonics	Name	Database	State	Result	Description		Start Item	Number of Items	Maximum of Items
Mars	<b>i</b> 3D		Fatai		PDF VDP template hasn't been identified.		1	0	0
Settings	i pattern	Counter	Failed		Database Empty(Database is not connected) ERROR: no database		5	0	0
Manual	i pattern	Counter	Failed		Database Empty(Database is not connected) ERROR: no database		1	0	0
	i original_vdp_snr	Counter	Failed		Database Empty(Database is not connected)		1	0	0
	i original	Counter	Failed		Database Empty(Database is not connected)		1	٥	0
	i original	Counter	Failed		Database Empty(Database is not connected)		1	0	0

## **Job Station**

The Job Station is where jobs to be executed are managed.

The **Job Station** shows 4 list of jobs:

- 1. Processing Jobs/Processed Jobs
- 2. Archived Jobs
- 3. Activity Log

In the **Job Station** you can:

- Create a new job.
- Manage existing jobs.

## Job management

The Job Station has 4 lists in which you can manage existing jobs and execute actions.

#### **Processing Jobs / Processed Jobs**

The Processing Jobs list contains the created jobs, the ones in preparation and the ones that are executing.

This Processed Jobs list contains the completed jobs: the executed ones and the aborted or failed ones.

#### **Archived Jobs**

This list contains the archived jobs. Jobs are listed here indefinitely until they are removed permanently.

#### **Activity Log**

This list contains ALL jobs: the processing ones, the processed and the rerun. For rerun jobs only the last run is listed. Individual listings can be deleted.

#### **Activity Details**

This pane reports the events and actions related to each run of the job.

To view the Activity Details pane, do the following:

- 1. Select a job.
- 2. Click the expand icon

A dialog shows you an read only preset.

## Actions

Element	Name	Description
Queue is PAUSED	Queue status: Paused or Running. Buttons: <b>Paused</b> or <b>Running</b>	When the queue is PAUSED, all jobs are processed up to the state "Pending".
		When the queue is RUNNING, the job on the top of the list starts when its state is "Ready to Execute". When this job is completed (fully executed or aborted) then the next job starts once that has the state "Ready to Execute"
8	Refresh	Allows you to reload the list view. When unused default refresh is 10 seconds.
* *	Move to the End / Move to the Top	Allows you to move the selected job to the bottom/top of the list.
~ ^	Move Down / Move Up	Allows you to move the selected job 1 row down/up in the list.
🗶 Abort	Abort	Allows you to abort the selected job. Aborted jobs are moved to the <b>Processed Jobs</b> list.
i	Info	Allows you to view a read only <b>Job</b> <b>Settings</b> window of the selected job while it is being processed. It also shows a preview of the uploaded image.
🚯 Rerun	Rerun	Allows you to rerun the selected job.
💼 Remove	Remove	Allows you to remove the selected job.
Archive	Archive	Allows you to archive the selected job.

## Filter jobs

The column headers represent various filter fields to find a job. They can be toggled to show/hide in the **Settings** on the right.

Name Database State Result Description Start Bern Number of Maximum of Berns

• Name

- Database
- State
- Result
- Description
- Start Item
- Number of Items
- Maximum of Items
- Creation date
- Modification date

Enter the full search term or part thereof to repopulate a column.

## Prioritize and pause jobs

If necessary, in the Jobs lists, you can select a job and do one of the following:

- **Prioritize** jobs by moving hem up/down the list by clicking  $\land$  and  $\lor$
- **Pause** a job by clicking the Queue is RUNNING button.
- View job specs by clicking the info icon **1**.

## CAUTION: Reorder Jobs

A job in the state "Ready to Execute" starts automatically if the queue is RUNNING and if there is no job above it in the list.

## Abort jobs

- Abort a job by clicking X Abort.
- Abort: interrupts any job at any time. Jobs aborted are moved back to the Processed Jobs queue.

## Delete jobs

A job can be deleted from Archived Jobs and Activity List. Deleting jobs is permanent.

Delete a job by clicking the **Remove** button **t** Remove

## Rerun jobs

A job can be rerun from Processed Jobs and Archived Jobs

- 1. Select a job.
- 2. Click the **Rerun** button 🚯 Rerun

A dialog shows you an editable Job Settings window.

3. If needed, modify the settings and click Submit.

## **Create a Job**

Upload or use PDFs from CLOUDFLOW and configure their imposition onto sheets and processing.

To create a new job in Job Station, follow these steps:

1. At the top right click the Create a Job... 🕒 button.

A job configuration window is displayed.

Create a Job		×
Start From Item	1	
Number of Output Items	0	
	Number of items to output or "0" for all from the start item.	
Copy Output File to		
Upload Cloudflow		
		*
Browse files		Ť
	Cancel Submit	

- Start from Item: defines from which record onwards you want to generate output files.
- **Number of Output Items**: defines how many records you want to output. If the number entered is bigger than the number of items (records in the database), then blank pages/documents will be generated.
- 2. Browse a file, or drag a file into the Upload pane.

Here you can define the VDP input file that was prepared in PACKZ and linked with variable elements. This can be a 1up file or a Step&Repeat file.

More than one file can uploaded. The minimum requirement is a PDF prepared in PACKZ. If this PDF is designed to get the Variable data from a CSV file, then a CSV file should be uploaded.

Optionally, additional files (PDF and/or images) can be uploaded. These Variable Data files can be used, for example, to change a background image.

All input files can be uploaded as a ZIP file. The ZIP file will be expanded automatically by the job.

**Note:** See the PACKZ manual for more information on how to Prepare VDP.

#### 3. Select Submit.

While they are being processed for imposition the job is sent to the **Processing Jobs** list. On completion the job is sent to the **Processed Jobs** list.

Successful jobs have a **Download** button and a link to an **Overview** in Proofscope.

# **APPENDIX**

# **PROOFSCOPE** Quick Start

This chapter provides information about the basic features in PROOFSCOPE.

It is intended for PROOFSCOPE users that do not use CLOUDFLOW.

The chapter content is designed to be copied to a text editor where it can be customized or refined if needed. When finished, it can be shared with PROOFSCOPE users.

If you want to extract a PDF from the chapter without editing it first, you can open it in the HTML version of the CLOUDFLOW User Manual and select 🖨 to print it to PDF.

The content will exist in several languages. Other languages besides English are to be added.

## **EN** version

PROOFSCOPE is an application for soft proofing.

## What can you do with PROOFSCOPE?

- View files in these formats:
  - PDF, JPEG, BMP, CF2 files.
  - 3D files.
  - 1 bit TIFF files and LEN files.
- Compare different files.
- Approve, reject or delegate files in case of an approval.

## **View PDF files**

With PROOFSCOPE you can view PDF files.

# ₽ 8 ✓	🏭 🗢 50% 🗢 183 と 🕸 👂 🕂 🖉 🖉	III admin
		Your decision is required: Show Approval Form
	est. 2015 PACKZ TEA quality Promium	est. Dest PACKZ TEA gaudity

- A: Sidebar.
- **B**: Toolbox.
- **C**: Download buttons.

Sidebar			
::	Thumbnails	Gives an overview of the pages. In case you're viewing a multipage file, all pages will be displayed.	
	Notes	<ul> <li>Gives an overview of the notes.</li> <li>Select </li> <li>to view notes per user.</li> <li>Select </li> <li>to view notes per page.</li> <li>Select Hide notes to hide all the notes on the design.</li> <li>Select Show notes to show the notes on the design.</li> <li>Select Show notes of previous versions to show the notes from older versions of the file.</li> </ul>	

	Sidebar	
R	Reports	Gives an overview of the preflight results of the <b>Analyze</b> node.
		Select a checkbox to view a specific type of preflight result. The problem area in the design is indicated by a red square.
		Select the triangle to expand. If you select a specific preflight result, the error becomes green transparent. When zoomed in, the zoom will automatically shift to the error area.

Sidebar			
	Separations	Gives information about the file. <b>SEPARATIONS</b> : the separations to display. Select <b>Edit</b> to edit the separations. You can	
		<ul> <li>Change the separation name.</li> <li>Change the spot colors.</li> <li>Change the opacity of the spot colors.</li> <li>Change the separation order by dragging and dropping the separations to a different order.</li> </ul> SEPARATION OPTIONS	
		<ul> <li>Single separation as black : checkbox to display the single separation in black. This option only makes sense if you have selected one separation in the SEPARATIONS list.</li> <li>Invert separation order: checkbox to invert the separation order. This option is only for previewing purposes and does not change the actual order of the separations.</li> </ul>	
		INSPECTION	
		<ul> <li>Show inspection overlay : option to check the design against specific characteristics.</li> <li>Minimum dot: a red overlay layer will highlight the areas in the design that have an ink percentage less than the minimum percentage predefined in the First visible dot field.</li> <li>Maximum dot: a red overlay layer will highlight the areas in the design that have an ink percentage higher than the maximum percentage predefined in the Maximum dot field.</li> <li>Total area coverage: a red overlay layer will highlight the areas in the design that have a total area coverage higher than the percentage defined in the Coverage field.</li> </ul>	
		<b>PAGE BOXES</b> : information about the page boxes.	
		<b>PAGE SIZE</b> : the page size of the design.	

	Sidebar			
✓	Approval Form	Displays the approval form that is created in the <b>Start Approval</b> node. In case one or more fields are required to be filled in, the user cannot approve or reject the file as long as these required fields are empty.		
		Note: The Approval Form is only visible when the checkbox <b>Display Form</b> in the <b>Start Approval</b> node is selected.		

	Toolbox			
:	Show Sidebar	Allows to open or close the <b>Sidebar</b> on the left.		
•	Zoom in	Allows to zoom in.		
100%	Zoom	Allows you to enter a zoom percentage.		
•	Zoom out	Allows to zoom out.		
	Fit	Fits the complete design in the window.		
5 0	Rotate	Rotates the design to the left or to the right.		
च् <b>म</b>	Mirror	Allows you to mirror the design.		
Q	Zoom	Allows you to magnify a selected part of the file. If you click, you zoom in, if you ALT + click, you zoom out.		
4	Pan	Allows to pan on the design while zoomed in. To pan, hold down the left mouse button and move the mouse. If you selected the <b>Thumbnails</b> view in the <b>Sidebar</b> , the zoom area is highlighted. To move the zoom area, hold down the left mouse button and drag it to another position.		

Toolbox			
₽.	Notes	Adds notes to the design. There are several types:	
		• Note creates a note on the location where you click in the design.	
		• Rectangle draws a rectangle around an area in the design.	
		• <b>Oval</b> draws an oval around an area in the design.	
		• Freehand draws a free form around an area in the design.	
		• <b>T</b> Note from Text draws a rectangle around existing live text in the design and copies the text to the text box.	
		After choosing the type, you can write your comment in the <b>Note</b> <b>window</b> . Select <b>Save</b> to save the note.	
		All notes are visible in the <b>Notes</b> on the <b>Sidebar</b> and on the design itself. Select <b>Hide notes</b> to hide them.	
		To open a note, click on the note in the <b>Sidebar</b> or by double click the visualization highlights in the design itself.	
Ø	Densitometer	Allows to measure separations.	
		Cyan 0% Magenta 14% Yellow 78% Black 0% PANTONE 384 C 0% PANTONE 871 C 0% Keyline 0% To measure separations, select the exact spot in the design that you want to measure.	

Toolbox	
Toolbox Measure	Allows to measure: • Vertical, horizontal and diagonal distances:
	the bottom left. Next, draw a circle in the file. The diameter of the circle corresponds with the diameter you have defined in the box. You can move the circle by dragging it to another position. Everything outside the circle has a bigger measure than the defined diameter. For example, this way you will know if a trap meets a

Toolbox			
Detect Barcodes	Recognizes and reads barcodes. To read the type and value of the barcode, drag a rectangle around the barcode while holding down the left mouse button. Barcode Detection Type EAN 13 Text 1234567890982		
	You can sele barcode to c software too	ect the value copy-paste it bl.	of the to another
	These are the supported barcodes:		barcodes:
	product	industrial	20
	UPC-A	Code 39	QR Code
	UPC-E	Code 128	Data Matrix
	EAN-8		
	EAN-13		
	1		

Download buttons			
Ð	Generate Report	Downloads an HTML PROOFSCOPE report.	
<b>F</b>	Download file	Downloads the file in low resolution.	

## View 3D files

With PROOFSCOPE you can view these types of 3D files:

- .ic3d files (IC3D).
- .dae files, which are XML files without graphics or other binary data (COLLADA).
- .zae files, which are compressed files containing the .dae file, a manifest file, graphics... (COLLADA).

Move around the 3D design by holding down the left mouse button.

## **View screened files**

With PROOFSCOPE you can view screened TIFF and LEN files after a RIP process.

Toolbox			
**	Measure Halftones	<ul> <li>Gives detailed information about the screens in a selected area:</li> <li>Separations</li> <li>Angles</li> <li>Resolutions</li> <li>Percentages</li> </ul>	
		To select an area, left-click or drag the cursor over an area while holding down the left mouse button.	



To view/measure an individual separation, deselect all separations except the one you want to view in the **Separations** view in the **Sidebar**.

#### Compare files

Icon	Description
A	Represents the first file
Δ	<b>Show difference</b> highlights the differences between the two files
В	Represents the second file
A/B	<b>Switching Files</b> automatically switches between file A and B
E3	Show Side by Side displays the files side by side

The comparison can be tweaked with the Differences view in the Sidebar (III).

## **COMPARE VERSIONS**

This option is only available in case you work with different versions of the same file.

- Compare: select this checkbox to compare different versions of the same file.
- Version A: the file that represents the first file (version A).
- Version B: the file that represents the second file (version B).

#### FILE B

- Offset X: the vertical offset (in mm) of file B with respect to file A.
- Offset Y: the horizontal offset (in mm) of file B with respect to file A.
- Rotation: the amount of rotation of file B with respect to file A. Options are 0°, 90°, 180° and 270°.
- **Reset**: reset the options to the default settings.

#### BOX FIT

• **Box**: the box that will serve as a reference point in case the files have a different position and you want to align them.

#### AUTOMATIC FIT

Use this option if you want to define a specific reference point on both A and B documents to align them.

- File A: the reference point on file A. Make sure file A is visible, select **choose**, position the cursor where you want to place the reference point on file A and click.
- File B: the reference point on file B. Make sure file B is visible, select **choose**, position the cursor where you want to place the reference point on file B and click. Next, select **Set** or **Calculate**.
- Set: the designs will be aligned exactly on the two reference points you have defined in File A and File B.
- **Calculate**: this button will use the reference point you have defined in File A and File B, but will use information of the images (such as colors) to align the designs more exactly.
- **Reset**: reset the options to the default settings.

#### ADVANCED

- Mode: if needed, select a different view mode to visualize the differences more clearly. Options:
  - **Default**: the default color mode will display the common pixels in File A and File B in white and the pixels that are not common in red. The pixels that are present in File A and not in File B are displayed in a darker red, the pixels that are present in File B and not in File A are displayed in a lighter red.
  - **Inverse**: the common pixels will be displayed in black and the pixels that are not common will be displayed in red. The pixels that are present in File A and not in File B are displayed in dark red, the pixels that are present in File B and not in File A are displayed in lighter red.
  - Hard: the common pixels will be displayed in gray. The pixels that are present in File A and not in File B will be displayed in blue, and the pixels that are present in File B and not in File A will be displayed in red.
  - Darken: both the common pixels and the pixels that are not common will be displayed in the original colors.
- Threshold: increase/decrease the threshold to see more/less differences

#### **Assess files**

In PROOFSCOPE you can approve, reject or delegate files in an approval workflow.

If a decision is required, a button **Show Approval Form** is displayed on the top right. If you select that button, the Approval Form is displayed in the Sidebar:



In the Approval Form, you can approve or reject the file.

If you have selected the checkbox **Allow Delegating** in the **Start Approval** node, a button **Delegate...** is also displayed. If you select this button, the decision to approve or reject the file will be delegated to another user. Options:

- Add approver: an extra participant of the approval workflow is added. This participant will receive a link to the file and will also be invited to approve, reject or delegate the file.
- Add informative approver: an extra participant of the approval workflow is added. This participant will receive a link to the file and will also be invited to approve, reject or delegate the file. However, this participant's decision is strictly informative and does not have an influence on the approval workflow.
- **Transfer approval**: the file is sent to another participant (who is not the initial approver). This participant will receive a link to the file and will also be invited to approve, reject or delegate the file.
- Approver e-mail: here you can enter the participant's email address.

#### **PROOFSCOPE** shortcuts

These are the shortcuts you can use in PROOFSCOPE:

	<b></b>	<i>R</i>
Fit zoom	Cmd 0	Ctrl 0
Zoom in	Cmd +	Ctrl +
Zoom out	Cmd -	Ctrl -
Zoom in when in zoom mode	Click	Click
Zoom out when in zoom mode	Option + click	Alt + click

# Data Structure of HelioDisk Job Details

This data describes all the information that can be specified in the CLOUDFLOW HelioDisk Job Details to generate a HelioDisk **.pro** file.

## HelioDisk Job Details

The following fields are defined in the CLOUDFLOW HelioDisk Job Details:

- **jobname** (*optional*, *string*): a name to assign to a job.
- printmaterial (optional, string): defines the material the job will be printed on.
- **comment** (*optional*, *string*): a comment to associate with the job.
- cylinders (*optional, array*): defines the parameters that are specific per cylinder that is part of the generated HelioDisk .pro file. See HelioDisk Cylinder below for a description of all possible fields.
  - **Note:** Each entry in this array must be linked to the correct plate (tiff file), and this can be either be done by index, or it can be done based on colorant names, and in that case the **colorant\_name** key is required and controls the linking.
- test\_cut (optional, object): defines test cut parameters. The following sub fields are defined:
  - layout (required, object): defines the test\_cut layout parameters. The following sub fields are defined:
    - origin (required, object): defines the origin of the test cut layout. The following sub fields are defined:
      - **x** (*required*, *number*): origin in x direction (in mm).
      - y (required, number): origin in y direction (in mm).
      - **x\_base** (*optional, string, enumeration*): specifies the reference for the x coordinate. Possible values:
        - engraving: (default)
        - cylinder:
      - **y\_base** (*optional, string, enumeration*): specifies the reference for the y coordinate. Possible values:
        - gbs: (default)
        - picture:
    - size (required, object): defines the size of the test cut layout. The following sub fields are defined:
      - width (required, number): width (in mm).
      - height (required, number): height (in mm).
    - **tile** (*required*, *object*): defines the size of a single tile in the test cut layout. The following sub fields are defined:
      - width (required, number): width (in mm).
      - height (required, number): height (in mm).
    - **line\_order** (*optional, string, enumeration*): the line order to use. Possible values:
      - axis: (*default*)
      - circumference:
    - line\_length (required, number): line length (in mm).

## HelioDisk Cylinder

This data describes the information that can be set per cylinder when generating a HelioDisk .pro file.

The following fields are defined in the CLOUDFLOW HelioDisk Cylinder data:

• **gradation** (*optional*, *string*): a URL to a gradation file. This can be an absolute URL to an actual gradation file, or it can be a relative URL (which means that a base URL for gradation files must be specified at the time of conversion to a .pro file), or just a file name (but still URL encoded).

- quality (optional, object): defines quality parameters. The following sub fields are defined:
  - **font\_width** (*optional, string*): a URL to a font-width (anti-aliasing) file. This can be an absolute URL to an actual gradation file, or it can be a relative URL (which means that a base URL for **font\_width** files must be specified at the time of conversion to a .pro file), or just a file name (but still URL encoded).
  - **engraving\_amplification** (*optional, string*): the name of a engraving-amplification file, URL encoded. You don't need to specify the extension since it will be appended by default when needed.
  - **sharpen** (*optional*, *object*): defines sharpen parameters. The following sub fields are defined:
    - **axial** (*optional*, *number*): defines axial sharpening. **1** based value. 1 is no sharpening.
  - circumference (optional, number): defines circumference sharpening. 1 based value. 1 is no sharpening.
- engraving (optional, object): defines engraving parameters. The following sub fields are defined:
  - method (optional, string, enumeration): the engraving method to use. Possible values:
    - normal:
    - laser:
    - extreme:
  - formation (optional, string, enumeration): the engraving formation to use. Possible values:
    - circular:
    - helix:
  - **enable** (*optional*, *boolean*): enables/disables engraving. Engraving will be enabled if this parameter is not specified.
- mirror (optional, object): defines mirror parameters. The following sub fields are defined:
  - circumference (optional, boolean): enables/disables mirroring in circumferential direction.
- screen (optional, object): defines screening parameters. The following sub fields are defined:
  - value (*optional, string*): the screening value.
  - **angle** (*optional*, *integer*): the screening angle.
- test\_cut (optional, object): defines test cut parameters. The following sub fields are defined:
  - data (optional, object): defines the test cut data. The following sub fields are defined:
    - **preset** (*optional*, *object*): a reference to a (CLOUDFLOW) CLOUDFLOW HelioDisk Job Details preset file. From this file, only the relevant **test\_cut** data section is extracted. All other parameters in this data section are ignored if a preset is defined. The following sub fields are defined:
      - **url** (*required*, *string*): a URL to the preset file.
    - values (*required, array of objects*): the values. Currently this array must always contain three entries. Each entries can contain the following sub fields:
      - input (required, integer): defines the digital input data. Value between (and including) 1 and 254.
      - transverse\_diagonal (required, integer): defines the transverse diagonal in micrometer.
      - **longitudinal\_diagonal** (*optional, integer*): defines the longitudinal diagonal in micrometer. Must be specified in the first entry, and optional for other entries (unless **true\_volume** is enabled).
    - canal (required, integer): defines the canal in micrometer.
    - **angle** (*required*, *integer*): the screening angle.
    - **true\_volume** (*optional*, *boolean*): enables/disables True Volume mode.
    - wall\_width\_correction (*optional, object*): specifies wall width correction parameters. By default wall width correction is disabled if this parameter is not specified.
      - **enable** (*optional, boolean*): enables/disables wall width correction. By default wall width correction is enabled if this key is not specified in the enclosing **wall\_width\_correction** parameter.
      - minimum (required, integer): defines the minimum wall width correction in micrometer.
      - maximum (required, integer): defines the maximum wall width correction in micrometer.
- **name** (*optional*, *string*): the name of the cylinder.
- length (*optional*, *number*): the length of the cylinder (in mm).

- id (optional, string): the ID to assign to the engraved cylinder.
- web\_shrink\_compensation (optional, number): a compensation factor for paper shrink while printing (in mm).
- gbs\_offset (optional, number): specifies the GBS offset (in mm).
- comment (optional, string): a comment to associate with this cylinder.
- colorant\_name (*optional, string*): the colorant name of this cylinder. This information is not really needed for the HelioDisk .pro file, but can be used to match cylinder info coming from two sources. For example, if have a set of files representing the plates and you also have some XML/JSON data coming from an MIS, you might need to match the plate names with the cylinder *array*. You can do this with the colorant\_name entry.

## Functions in the Script node

You can use various functions in the Script node.

- The standard JavaScript functions and constructs.
- The API functions (the range depends on your license).
- Additional functions listed below:

#### getParameters()

This function obtains the parameters that were specified for the Script node.

This will give you access to the variables of the workable, the files you need to process, etc.

#### Example

getParameters()

#### getJSONFromContainer()

This function obtains JSON data from a container in the database.

In most cases, this data originates from work data that was saved by a previous Data or Script node. In those cases you have a reference to that data, and this call will load the data into the script. The data\_reference parameter can be an object containing a **data provider value composer** node, or a string. The advantage of using the object is that the references stored in the workable are typically inside objects and as such they can be used **as is**.

#### Example

```
getJSONFromContainer({\"type\" : \"com.nixps.quantum.data_provider.0\",
  \"url\" : \"cfqworkdata:///1234/\"}
getJSONFromContainer(\"cfqworkdata:///1234/\"})
```

#### setResultFiles(files)

This function sets the file reference(s) that should be assigned as result of this node. The files parameter can be a single string for a single file, or an array of strings for multiple output files.

#### Example

```
setResultFiles("cloudflow://PP_FILE_STORE/MyFile.txt");
setResultFiles(["cloudflow://PP_FILE_STORE/file1.txt" , "cloudflow://
PP_FILE_STORE/file1.txt"]);
```

#### setResultContainerFromJSON(json\_data)

This function saves the specified JSON data in the workable data and adds a reference to the saved JSON data and to the result of this node. The **json\_data** parameter will be saved **as is** for later use.

#### Example

setResultContainerFromJSON({"key1" : "value 1", "key2", "value 2"});

#### setResultVariables(variables)

This function sets the variables that should be assigned as a result of this node.

#### Example

setResultVariables({"var1" : 1, "var2" : "Some Text"});



**Note:** CLOUDFLOW checks the names and values of the specified variables to make sure they comply to the variable specifications.

#### setOutput(connectorName)

This function controls the output path of the workable when the script is finished. For example, when you specify **failure**, the workable will be sent to the failure output.

#### Example

```
setOutput('failure');
```

#### console.log(...)

This function adds the specified **text** to the workable log. The parameters can be text or complete JSON objects that are *stringified*. Multiple arguments are space separated.

The main differences between this function and the **print()** function are the following:

print()	console.log()
Can handle one argument	Can handle 0, 1 or multiple arguments
Encloses a string in double quotes	Does not quote strings

console.log() is API compatible with console.log in the browser.

#### Example

```
console.log('nr seps', 2);
```

#### addMessageToLog(severity, message)

This function adds a message to the workable log. The difference between this function and the **print()** function is the following:

print()	addMessageToLog(severity, message)
Is always <b>info</b>	Possible to specify a <b>severity</b>

Possible severities are:

- debug
- info
- warning
- error

#### Example

```
addMessageToLog('info', 'some text');
```

#### setProgress(value, message)

This function sets progress info. This contains a progress value between 0 and 1 and activity information. This progress and activity will be fed back to the dashboard (when the script takes a long time).

#### Example

```
setProgress(.5, 'working');
```

#### addSystemMessage(severity, message)

This function adds a message to the system log.

Possible severities are:

- debug
- info
- warning
- error

#### Example

```
addSystemMessage('warning', 'hey, you need to know this');
```

**Note:** The **addMessageToLog** function sends the feedback to the workable log, and sometimes to the system log.

#### addEventMessage(severity, message, url)

This function adds an event message to the system log linked to the specified asset. The event type will be **kGenericAssetMessage**.

Possible severities are:

- debug
- info
- warning
- error

#### Example

```
addEventMessage('error' , 'asset is broken', 'cloud flow://PP_FILE_STORE/
test/asset.pdf');
```

## Using custom objects in the database

With custom objects you can store custom data in MongoDB.

Custom objects are stored in collections with the prefix **customobjects**. The collections are created automatically when a custom object is generated.

Custom objects are scoped. This means that if a scope is assigned to a user, this user can only see and manipulate objects to which the same scope has been assigned. Other objects are not visible to the user. To users to which all scopes are assigned, all objects are visible and editable.

Some collections may need to contain publicly accessible objects. You can do this by setting a preference that disables the scope checking for a particular **customobjects** collection. For a collection called **my\_collection**, you can disable the scope filtering with this command:

```
api.preferences.save_for_realm(false, 'system', '',
    'com.nixps.customobjects.my_collection, 'scoped');
```

You can reactivate the scope filtering by replacing false with true.

## **Creating objects**

An object can be created using custom\_objects.create, which takes the name of the collection and the new object as parameters.

#### Example

```
obj = api.custom objects.create('my collection', {foo: "test", bar: 1});
```

After this call:

- **obj** is a copy of the object.
- \_id is the extra key, which is the object ID.

#### Note:

- If a single scope is assigned to the user, the same scope is assigned to the object. Consequently, the object is only visible to users to whom the scope is assigned and to users to whom all scopes are assigned.
- If all scopes are assigned to the user, no scope is assigned to the object. Consequently, the object only is visible to users to whom all scopes are assigned.

If all scopes are assigned to the user, the user can assign a scope to the object using custom objects.set scope.

#### Example

```
api.custom objects.set scope('my collection', objectID, scopeID);
```

Where:

- **objectID** is the \_id field of the object (assigned by the create function).
- **scopeID** is the Mongo ID of the scope.

To remove the scope of the object, pass an empty string as the scopeID.

#### **Retrieving objects**

If the Mongo ID of the object is known, the object can be retrieved using custom objects.get

#### Example

obj = api.custom\_objects.get('my\_collection', objectID);

A custom object collection can be queried using custom\_objects.list and custom objects.list with options:

```
api.custom_objects.list('my_collection', ['foo', 'equal to', 'test'],
    ['foo', 'bar']);
```

```
api.custom_objects.list_with_objects('my_collection', ['foo', 'equal to',
    'test'], ['bar', 'ascending'], ['foo', 'bar'], {maximum: 5})
```

#### **Changing objects**

There are three ways to change a custom object:

1. The function custom objects.update replaces the object with another object:

api.custom objects.update('my collection', objectID, {x: 1, y: 2});

2. The function custom objects.set keys performs a partial update.

#### Example

To replace the **bar'** field and to add the **xyz** field:

```
api.custom_objects.set_keys('my_collection', objectID, {bar: 13, xyz:
"abc"});
```

3. The function custom\_objects.remove\_keys removes some fields from an object:

```
api.custom objects.remove keys('my collection', objectID, ['foo']);
```

#### Removing objects

An object can be removed using the custom objects.delete function:

```
api.custom objects.delete('my collection', objectID)
```

The function custom objects.delete many removes several objects at once:

```
api.custom_objects.delete_many('my_collection', ['foo', 'equal to',
    'test']);
```

## **Output File Path Generation**

In several nodes (for example, the **CONVERSION** nodes), CLOUDFLOW needs to generate an output file. This chapter describes the rules how the output file is generated.

Take a look at this example. The **Image to PDF** node has access to an input file and contains a parameter to specify an output path:

Image To PDF	8	×
Image File:		
All files from node previous Node	, er	
Stacked Files:		
PDF File:		
Lossless compression:		
Overwrite existing file:		
Close	Sav	e
In this case, the generation of the output file follows the standard input-output file name combination rules. These are the following:

If	Then	
The output is left blank	The output file will be written in the same folder as the input. Depending on the type of conversion, the result can be:	
	<ul><li>A file name that has a suffix appended, with the original extension.</li><li>The original file name with a different extension.</li></ul>	
The output is set to an extension	The output file will be written in the same folder as the input. Depending on the type of conversion, the result can be:	
	<ul><li>A file name that has a suffix appended, with the extension you specified.</li><li>The original file name with a different extension.</li></ul>	
The output is set to a file name without extension	The output file will be written in the same folder as the input. Depending on the type of conversion, the result can be:	
	<ul><li>The specified file name with the original extension.</li><li>The specified file name with a different extension.</li></ul>	
The output is set to a file name with extension	The output file will be written in the same folder as the input, with the specified file name and the specified extension.	
The output is set to an absolute path	The output file will be written in the same folder as the input. Depending on the type of conversion, the result can be:	
	<ul><li>A file name that has a suffix appended, with the original extension.</li><li>The original file name with a different extension.</li></ul>	
The output is set to a relative path	The output file will be written in a folder based on the original input folder, appended with the specified relative path. Depending on the type of conversion, the result can be:	
	<ul><li>A file name that has a suffix appended, with the original extension.</li><li>The original file name, but with a different extension.</li></ul>	
The output is set to an absolute or relative path, in combination with file name and/or extension	The combination rules for the folders, file name and extension as described above will all be applied to generate an output file name.	

#### 

- Note:
- If you do specify a full absolute path, the output file path will be exactly that path in all cases.
- If you only specify a path (relative or absolute), it must end with a slash.

#### Example

Output Parameter	resulting Output Path (for conversion)	resulting Output Path (for update)
	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/file.pdf	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/file_alt.txt
.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/file.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/file_alt.pdfa
written	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/written.pdf	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/written.txt
written.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/written.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/written.pdfa
cloudflow://PP_FILE_STORE/ Folder3/	cloudflow:// PP_FILE_STORE/Folder3/ file.pdfa	cloudflow:// PP_FILE_STORE/Folder3/ file_alt.txt
Folder4/	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/Folder4/file.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder2/Folder4/ file_alt.txt
/Folder4/	cloudflow:// PP_FILE_STORE/Folder1/ Folder4/file.pdfa	cloudflow:// PP_FILE_STORE/Folder1/ Folder4/file_alt.txt
cloudflow:// PP_FILE_STORE/Folder3/ written.pdfa	cloudflow:// PP_FILE_STORE/Folder3/ written.pdfa	cloudflow:// PP_FILE_STORE/Folder3/ written.pdfa

The input is **cloudflow:**//**PP\_FILE\_STORE**/**Folder1**/**Folder2**/**file.txt**. In case of both a conversion from txt to pdf and an update that will write an alternative text file, the results are the following:

# GLOSSARY

# Anti-aliasing

Anti-aliasing is used to smoothen aliasing (jaggies and stairstep-like lines) effects. Aliasing occurs because the output device, the monitor or printer doesn't have a high enough resolution to represent a smooth line.

# Asset

An asset is file information stored in a database. Each asset has a unique CLOUDFLOW URL to reference it.

The difference between an asset and a file is that an asset refers to information in a database, while a file refers to a physical file (on a disk).

Assets are located in File stores.

# **Bar Width Reduction**

When a file with a barcode is printed, the barcode bars will usually print wider than the bars in the original document. This is called the ink gain or ink spread. Several factors can have an influence on ink gain: print pressure, substrate, ink type,...

To compensate for this, a Bar Width Reduction is applied when creating the barcode, where the value is divided over both directions of the bar width. A positive value will make the bars thinner; a negative value will make the bars thicker.

Bar Width Reduction is particularly interesting in the gravure printing industry.

### Bleed

Bleed is the area beyond the actual finished dimensions of a design. It avoids white borders around the label or package when it is printed and finished. The Bleed Box defines the bounding box of the design, including the bleed.

### **Blend Mode**

A Blend Mode determines how to objects are blended into each other. The object containing the blend mode is called the **blend object**, the underlying objects are called the **base objects**. The result is called the **blend color**.

There are two types of blend mode

- Separable blend modes: the term separable means that the blend mode is applied on each separation of the blend object towards the base object. In a separated color space such as CMYK, such blend modes are calculated on all 4 separations.
- Non-separable blend modes: the term non-separable means that the all colors are considered as one component to calculate the blend color.

Non-separable blend modes are **Hue**, **Saturation**, **Color** and **Luminosity**. All others are separable ones. The formulas of Non-separable blend modes are based on HSL conversions, meaning that the CMYK components are converted to their complementary RGB components, the formula is applied and converted back to CMYK. This can result in additional CMY or K separations even if they are not used in the document.

### CIP3

CIP3 stands for **International Cooperation for Integration of Prepress, Press and Postpress**. The result of this cooperation is the Print Product Format (PPF). The CIP3 Print Production Format is a container format for the exchange of data coming from prepress, press and postpress processes.

The information stored in a CIP3 PPF file can contain the following:

- administrative data (for example application name, copyright, ...)
- preview images for each separation that enables ink key presetting for the press
- transfer functions
- color and density measuring information
- register marks
- cut data
- folding procedures

# **Combined approval**

In case of a combined approval, an item needs several combined regular approvals and the final accept or reject will be based on the combination of the results of all these regular approvals. Users that are specified by **user**, users that are specified by **e-mail** and users that are specified by **attribute** will be combined so that every user will only appear once in the final participant list. Users that have a double definition (for example because they you specified them explicitly by e-mail but they were also included in the group that you selected by attribute) will only be included once. A user that appears once as a regular participant and once as an informative participant will be added to the list as a regular participant.

#### **Related information**

Regular approval on page 474

### Curves

Curves allow adjusting the color of each object or the entire design meeting the quality expectations of the printed result. It is an independent solution that fits in all RIP solutions.

Curves can be used to

- · compensate for the dot gain on a printing press
- · control the minimum and maximum tonal range in certain printing processes

# Distortion

In some printing processes, the substrate and/or plates might experience a certain amount of stretching. To compensate for this, horizontal and vertical distortion is applied.

### **Glossary term**

A glossary term is a word or a compound words has a specific meaning in a specific context.

### **Illustrator blend**

An Illustrator blend is a group of objects created by the **Illustrator Blend Tool**, in which one object is transformed to another in order to create morphed artwork or to create shaded objects.

### Jacket

A jacket combines several workables into a single logical unit.

At least one workable is created when starting a job. In some cases however, one workable is branched into several workflows which will independently follow the workflow. A good example is the approval workflow. A single approval workable will branch to multiple workables: one for each person to approve.

A jacket combines those several workables into one unit. A jacket also contains the full history of all referenced workables, and is kept forever. Each workable has always exactly one jacket, while a jacket can reference multiple workables.

You can visualize jackets and workables of a workflow in FLOWS > WORKABLES

**Related information** Workable on page 474

### mongod

mongod is the primary daemon process for the MongoDB system. It handles data requests, manages data access, and performs background management operations.

See this link for more information.

# Oversampling

Oversampling is important when converting linework to images. Without oversampling, jaggy or stair-stepped defects may appear along diagonal edges.

Oversampling generates more image pixels than it needs in the final image. It uses a higher resolution than the final image, which results in a smoother edge.

# Page boxes

The exact size of a PDF page is not as straightforward as it seems because there might be up to five different descriptions in a PDF that relates to its size. These are called **page boxes**.

- Media Box: this is the largest page box in a PDF. The other page boxes can be equal to the Media Box but they should not be bigger. In prepress use, pages are defined slightly oversized so that the bleed, the crop marks, information panels and other useful information are visible as well. This means that PDF documents used in graphic arts usually have a Media Box, which is bigger than the trimmed page size.
- Crop Box: this is the region to which the page contents are to be clipped. Adobe Acrobat uses this size for screen display and printing. For prepress use, the Crop Box is irrelevant.
- **Bleed Box**: this box determines the region to which the page contents needs to be clipped when output in a production environment. Usually the Bleed Box is 3 to 5 millimeters bigger than the Trim Box. Most prepress systems allow you to define the amount of bleed and ignore the Bleed Box.
- Trim Box: this defines the intended dimensions of the finished page. Unlike the Crop Box, the Trim Box is very important because it defines the actual page size. Most imposition programs and workflows use the Trim Box as the base for positioning pages on a press sheet or labels and packaging on a step and repeat.
- Art Box: the Art Box can define a region within a page that is of special interest. It is rarely used by applications.

### **Pull Back**

Pull Backs are often called **stay-away-traps**, **cutbacks** or **keepaways**. Pull Backs are needed when objects, in rich black or rich spot color, are positioned on light backgrounds.

- Rich Black is a composition of 100% black with a percentage of another separation. The result is a deeper look, making it more opaque.
- Rich Spot Color is a composition of a dominating separation containing a percentage of any other separation. The result is a deeper and vivid color.

For instance:

- gold with yellow undercolor
- a green spot separation with an additional green separation.

In both cases the lightest separations are pulled back to prevent revealing the lightest separations at the edges of the rich black or rich spot color, in case of registration errors.

### **Regular approval**

In case of a regular approval, an item needs to be approved by the list of specified participants. The item will be accepted or rejected based on the decision(s) of the participants.

#### **Related information**

Combined approval on page 472

# Screening

Screening is a reprographic technique that simulates continuous tone imagery using dots. Many screening technologies are controlled by a dot shape, angle and frequency, specified in a default set or for each separation separately.

### Strip

A baseline TIFF image is composed of one or more strips. A strip or band is a subsection of the image composed of one or more horizontal rows of pixels. Each strip may be compressed independently of the entire image, where each begins on a byte boundary. If the image height is not evenly divisible by the number of rows in the strip, the last strip may contain fewer rows. If strip definition tags are omitted, the image is assumed to contain a single strip.

In case of a single-strip TIFF file, a bitmap is stored in one large block. In case of a multi-strip TIFF file, the horizontal blocks of the image are stored together.

### Tile

Tiles are visual parts of a file preview in **PROOFSCOPE**.

# Trapping

Trapping is the process of creating slight overlaps between abutting colored objects within a document. It compensates registration errors between printing plates and printing towers to avoid potential gaps between colors in the design.

In general, light colored objects spread or expand into dark colored objects, as this is less noticeable for the eye than changes in light objects. The opposite can also happen: darker colored objects shrink into light colored object. This is called a choke and generally applied when the darker colored object is on top of the light one.

A color is defined darker or lighter than another color depending on the luminance or Lab value.

### Workable

A workable is a unit of processing, a single piece that travels through the workflow. It contains information about the files/data to operate on, the actions that already have been executed, the variables that can be used (for example trapping distance provided by the user during submit)...

When submitting an item to the workflow, one or more workables are created, traveling through the workflow, visiting the corresponding while following the connections you have defined. On correct processing, the success output will be followed; on failure, the failure output will be followed.

#### Example

In the following workflow, the path of the workable is indicated in blue:



You can visualize jackets and workables of a workflow in FLOWS > WORKABLES

#### **Related information**

Jacket on page 472

### White Underprint

White underprint is the technique of covering a transparent or metal substrate with white opaque ink before printing. This ensures that the substrate's nature does not interfere with the artwork when printing.

# **Regular expressions**

#### What is a regular expression

A regular expression is a sequence of characters that define a search pattern, where each character has a specific meaning.

For example  $b[A-Z0-9._{+-}]+Q[A-Z0-9.-]+$ .  $[A-Z] \{2, \}$  b defines any email address.

#### Using regular expressions in CLOUDFLOW

There are various functionalities in CLOUDFLOW where you can use regular expressions. Here are some examples:

• In SHARE, you can define a regular expression filter when creating a syncspec.

- In WORK SERVERS > FILE INDEXING, you can define a regular expression when defining an Asset Filter.
- In Form Builder > Item Repeater and Form Builder > Text you can define a regular expression to force the user to use a specific pattern for the text.
- In USERS > MANAGE SCOPES, you can define a regular expression to create a filter.

#### **Examples**

- .\*/Artwork/.\*\.pdf\$ matches all files with extension **pdf** that are located in the folder **Artwork**.
- ^PP FILE STORE/blech matches all files in a folder that starts with PP\_FILE\_STORE.

#### **Useful websites**

- On https://www.regular-expressions.info you can find a complete tutorial on regular expressions.
- On https://regex101.com you can online test regular expressions.

# Index

### **Special Characters**

.war file 44 <Patterning 445

### A

Actions PAGEBUILDER How to create an Action 105 Overview 105 What are Pagebuilder Actions 104 Active user 37 Add Worker 445 Add workflow 119 agnostic 443 Algorithms 444 Analyze file 174 aniloxes 443 Approval Combined 472 Regular 474 Approval assessment 161 Approval templates 301 Approve a file In KIOSK 300 In PROOFSCOPE 300 Art box 199, 473 Asset Metadata Overview 60 Search 60 Auto Update 119

# B

Back up MongoDB 36 Backup File stores 15 MongoDB 15 Barcode Bar Width Reduction 179 Recognize 179 Update 179 barcodes 443 Benefits 443 Bleed box 199, 473 Build a workflow 117, 131, 446 Build an approval cycle 299 Building a workflow Advanced 131 Routine 117

### С

Check License 445 CIP 184 CIP3 471 cloud 12 CLOUDFLOW Anti-virus 11 Firewall 11 Installation 18, 18 System requirements 10 Updating 22 CLOUDFLOW configuration for replication sets 25 CLOUDFLOW path 63 CLOUDFLOW SHARE Add a site 364 Create a sync specification 365 Define a site 364 Set up a SHARE network 364 Status 366 What is it 363 CLOUDFLOW sidebar 55 **CLOUDFLOW URL 63** Color book 189 Combined approval 299 Configuration 445 Contact Add Automatically 282 Manually 282 Edit 282 Container 239 Convert to Image 182 Copy file 153 Copy folder 154 Create folder 154 Creator 199 Crop box 199, 473 Curve Apply curve 199 Set page curve 199 Curves Curve Collections 356 Curve Details 361 Curve Profiles 357 Curve Recipes 357 Custom Objects Scope 466

### D

Data Versus files 239 Data connector Call custom function on 240 Get record from 241 List records from 242 Update record to 244 Data Locality 12 Database Add 370 Collections 52

JSON object 8 Maintenance 55 Records 52 Database collection 8 deep zoom 45 degradation 443 Delete file 155 Delete folder 155 Delete jacket 144 Demo 446 Deselect Fill 203 Font 203 Object 203 Stroke 203 Transparency 203 Distortion 199, 472 Dot gain 472 Dynamic content Approval overview 301 Notes overview 301 Variable 301 Workable messages 301

# E

End 144 EU 1169/2011 Violations 174 Export as CIP 184 as Image 184 Gravure 185 Export workflow 119 Expression builder 291

### F

Fail-over 25 File Search 60 File access restriction 286 File Info 155 File information 159 File management 60 File storage 38 File storage on a network share 38 File store Add file 60 Context Menu 60 Copy file 60 Cut file 60 Delete file 60 Download file 60 Overview 60 Rename file 60 Set up 38 Upload file 60 Flatten Image 187 Overprint 187 Shading 187 Flexo printing 443

Flexography 443 Force an approval 300 Form 139, 144 Form Builder How to use it 260 What is it used for 260

### G

Gravure Export 185

### H

hook 117 hooking 117 Hooks 131 Hot folder 139 HTTP 151 HTTP Service 133 HXM screening combinations 340 HYBRID 445

# I

Illustrator blend 472 Illustrator Blends 174 Image Convert to 182 Export as 184 JPEG Compression 184 Noise 182 Oversampling 182 Resolution 182, 184 Import workflow 119 Input/Output 445 Installation As a part of CLOUDFLOW 32 CLOUDFLOW 18, 18 MongoDB 18, 31, 32, 32, 32 On a separate machine 32 Intelligent Flexo 443 Invisible paints 174

### J

Jacket 472 Jackets Delete 124 Download 124 Java web host 370 JavaScript 133 Job configuration 450 create 450 Create 373 Job management 372 What is it? 372 Jobs KIOSK 382 processed 450 processing 450 JOBS API 372 How does it work? 372 Use case 383 What do you use it for? 372 Join PDF pages 156 JS API 391 JSON Update 243 JSON over HTTP 390 JSON syntax 8 JSON-RPC Request 390 Response 390

### K

Kiosk 145, 151 KIOSK Access 260 Jackets overview 258 Submit to workflow 258 UI 258 KIOSK parameters 140 KIOSK Parameters 260

### L

Layer Activate 192 Add 192 Deselect 192 Move 192 Remove 192 Rename 192 Select 192 Len To TIFF 444, 445 License 445 Live Objects Create 193 Template 193 Local file storage 38

### M

MARS app types Apps Shared With Me 403 My Apps 402 Public Apps 401 Media box 199, 473 Merge TIFF files 445 Meta data 164, 165 Modifier 291 MOM XML file 326 MongoDB Anti-virus 32 As a part of CLOUDFLOW 32 Back up 36 Custom objects 466 Firewall 32

Installation Linux 35 Mac 33 Windows 33 On a separate machine 32 Repair 36 Setting up the environment Windows 33 System requirements 32 Updating to WiredTiger 35 WiredTiger 32 Mounting method Automatic 326 Mirror mount 326 Move file 156 Move folder 157 Multiple CLOUDSERVER setup 25 Multiple participants 299

### Ν

Node Category 133 Configuration 117, 446 Connect 117 Copy-paste 117 Delete 117, 446 Disconnect 117 Drag to the workflow area 117, 446 Input 117 Label 117 Output 117 Overview 133 Reference 133 Settings 224, 445 Non-embedded fonts 174 nucleusd Commands 21

### 0

Open Paths 174 OpenGL 3.2 11 Out of sync Image 174 Station 174 Overprint Mode 1 174 Oversampling 473

### P

Page box 199 Page boxes 199, 473 PAGEBUILDER Activate 67 Areas 68 Build up the page 71 Components overview 72 Create page 67, 67 Edit an existing page 67 Translation 69 Use case 115

Pagebuilder components Activate 95, 96 Button 94 Chart 97 CLOUDFLOW topbar 72 Custom Grid 99 Custom page component 104 Deactivate 95, 96 Extended text editing 74 Form 93 Image 74 Job Table 86 Link 75 Navigation 75 Table 76 Tabs 92 Text panel 74 Title 74 Title bar 72 Twitter 96 YouTube 95 PATCHPLANNER job Create 326 Job settings 326 PATCHPLANNER sheet Create 329 Path Coordinates 191 Insert 191 Pattern 224 Pattern image 445 Patterning 224 PDF 156, 159 PDF/X-4 315 PEM certificate 23 Performance Tuning 47 PHP API 391 Pitstop 158 portal.cgi 390 Prerequisites 444 Print Product Format (PPF) 471 Print Production Format File (PPF) 184 Process 444, 444 Processed Jobs 450 Processing Jobs 450 PROOFSCOPE 3D 310 Approve 313 COLLADA 310 Compare files 311 Deep zoom 306 Delegate 313 IC3D 310 Inspect 304 Maximum dot 304 Measure halftones 310 Minimum dot 304 Notes 306 Reject 313 Report 309 SDK 316 Shortcuts 315

Sidebar 304 Supported formats 315 Toolbox 306 Total Area Coverage 304 View file 304, 304 View halftones 310 View page boxes 304 View separations 304 View TIFF file 310 PROOFSCOPE SDK Authentication 316 **CLOUDFLOW API 316** Component 323 Create difference URL 318 Create view URL 318 Customize user preferences 319 Get file metadata 317 Import file 316 Remove file 317 Retrieve asset thumbnail 319 Pull Back 210, 473

### R

Reduce Linework 205 Redundant clips 174 Registration errors 474 Remove workflow 119 Repair MongoDB 36 Repeat 148 Repetition 195, 206 Replication sets 25 Requirements 444 Resolution High 174 Low 174 **REST 226** RIP Calibration 223, 355 CIP3 223, 355 Color Management 223, 351 Color profile 363 Finished job 350 Input file 219 Job overview 349 Jobs window 349 Logs 363 Output 219, 350 Post Process 352 Presets 350 Separations 220, 352 Standalone 330, 349 Start a job 349 TIFF viewer 363 Web-based UI 349 Workflow 330 Routing 149

#### S

Save Incremental save 199 Save edit history 199 to Device 202 Scope Add 286 Assign 287 Create 286 Delete 286 Edit 286 Workable 288 Scopes Guest scope 287 Restrict access to workflows 288 Versus permissions 286 Screening 199, 444, 474 Screens AM 330 FM 338 HDS 338 HPS 330 HXM 340 Hybrid 340 Stochastic 338 Script 150 Search Asset 60 File 60 Select Fill 203 Font 203 Object 203 Stroke 203 Transparency 203 Select file 159 Select workflow 119 Separations Remap 189 Remove 189 Site CLOUDFLOW SHARE 364, 364 **SOAP 227** Split PDF pages 159 SQL query 371 SOL records 240 SSL 23 Start a new workflow 119 Start approval 161 Station Replace 201 Step and repeat Nested 195 Tabular 206 Step and repeat station 195, 206 sub flow 117 Sub flow 143, 147, 152 Submit to workflow 119 substrates 443 Sync specification **CLOUDFLOW SHARE 365** 

Synchronize 238

#### Т

Tag 164, 164, 165 Template based workflow 117 Text Replace 209 Search 209 TIFF 444, 444, 445 Timer 142 Tonal range maximum 472 minimum 472 Transform Delete 210 Mirror 210 Move 210 Rotate 210 Scale 210 Trapping 210, 474 Trim box 199, 473

#### U

Unzip 159 Updating CLOUDFLOW 22 URL 152 URL parameters 260 User Add 283 Delete 283 Edit 283 Permissions 283 User information 144 User Preferences 59

#### V

Variable 152, 289, 291 Variables Define 290 Specifications 291 Vectorize Bitmaps 205 Images 205 Opacity 205 Patterns 205 Strokes 205 Text 205 Virtual machine 12

#### W

Warp 217 Warping 217 Welcome page 286 White Paper 144 WiredTiger 35 Work server Add 40 Add Workers 41 Map File store 41 Workable 143, 474 WORKABLES Debugging 124 Delete 124 Download 124 State 124 worker 444 Worker Change default setup 47 Dataroom 42 Default setup 46 Event handler 42 File indexing 42 Garbage collector 43 Intelligent Flexo 43 Java web apps host 44 Metadata generator 44 Notification Worker 44 Overview 42 PACKZflow 44 Postgres 44 Preview generator 45 PROOFSCOPE Live Renderer 45 PROOFSCOPE renderer 45 Quantum combined 45 Quantum data 45 Quantum java 46 RIP 46 SHARE 46 SHARE Scheduler 46 Software update 46 Status 37 Tectonics 46 VDP Engine 46 Workflow Configuration 446 Start 446 Workflow area 119 Workflow Editor Shortcuts 124 Workflow hooks 131 Workflow list 119

# X

XML Create 241 Transform 242 Update 245 XPath 245

### Y

Yellow screen ruling 330

### Z

Zip 160