

Subflow

Description

Subflow is the easiest way to create "subflows".
This app can be useful whenever you need to integrate a flow within another flow.



This is a bundle composed of three elements:

- **Subflow - Main**, which sends a job to another selected flow
- **Subflow - Receive**, the input of the subflow, which receives jobs from "Subflow - Main"
- **Subflow - Send back**, one of the outputs of the subflow, which sends jobs back to their corresponding "Subflow - Main"

Compatibility

Switch 2022 fall or higher.

Connections

Subflow - Main

At least one incoming connection.
Traffic light outgoing connections.

Subflow - Receive

No incoming connection.
One output connection.

Subflow - Send back

At least one incoming connection.
No output connection.

Use cases

- Split a big workflow into smaller logical parts.
- Create some generic flows you can call into any other flow.

 **Subflow - Main**

This element is the one you use in your “main flow” to send incoming jobs to a “subflow”.

Flow element properties

Subflow

Select a "subflow" in which incoming jobs should be sent.

Use select from library property to show flow with receive element.

Subflow parameters

Optionally, you can specify "parameters" that may be used in the "subflow".

Those parameters will be set as private data keys.

Just set them as you would do with private data keys in a flow folder.

e.g.:

param1=value1

param2=value2

Timeout

Amount of time (in minutes) after which an incoming job will fail if it has not been sent to the corresponding "Subflow - Receive" element.

The job will be retried every 15s.

A value of 0 means this option is disabled.



Subflow - Receive

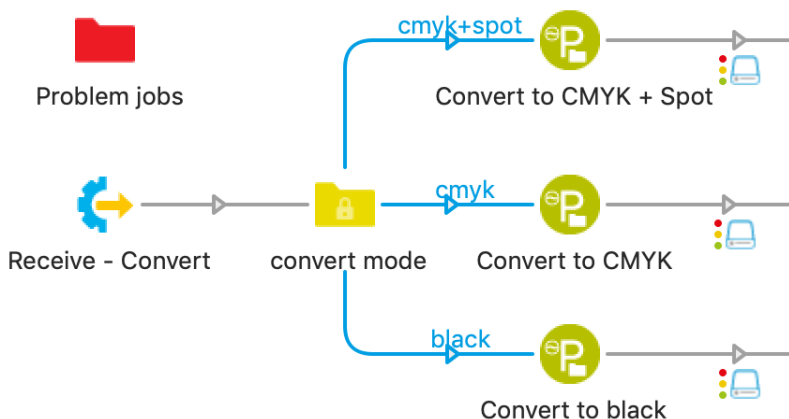
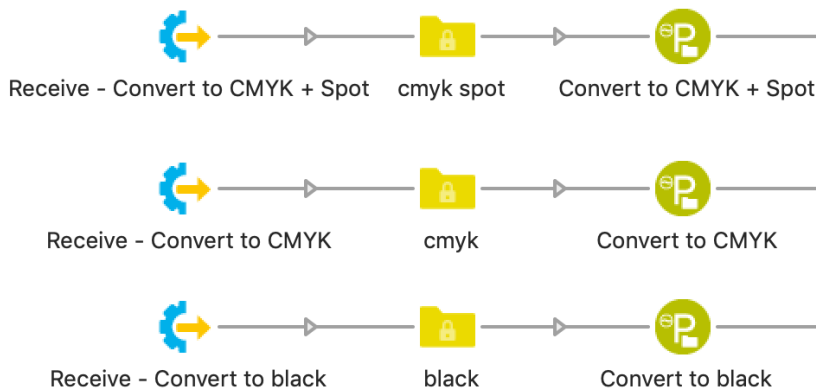
This element is the input of your "subflow". Just use one by flow.

This element has no property.

Disclaimer :

If several "Subflow - Receive" items are present in the same flow, the job will be randomly routed to one of them.

It is therefore preferable to use one "Subflow - Receive" element per flow



 **Subflow - Send back**

This element is the output of your "subflow". You can use several in your flow.

Flow element properties

Output level

Specify an output level for incoming jobs.

For **Success**, **Warning**, **Error** values, the output will be routed to the appropriate "Subflow - Main" element output level.

For **Fail** value, the job will be routed to the Problem jobs folder of the main flow.

Fail message

Value of the fail message for the main flow.

Default value is [Job.FailElement] : [Job.FailMessage]

Write log

If you want to output a log in the "Subflow - Main" element.

Log content

Enter the log content.

Log type

Choose a log type.

Possible values:

- XML
- Opaque

Log extension

Specify an extension for the generated log file.

Timeout

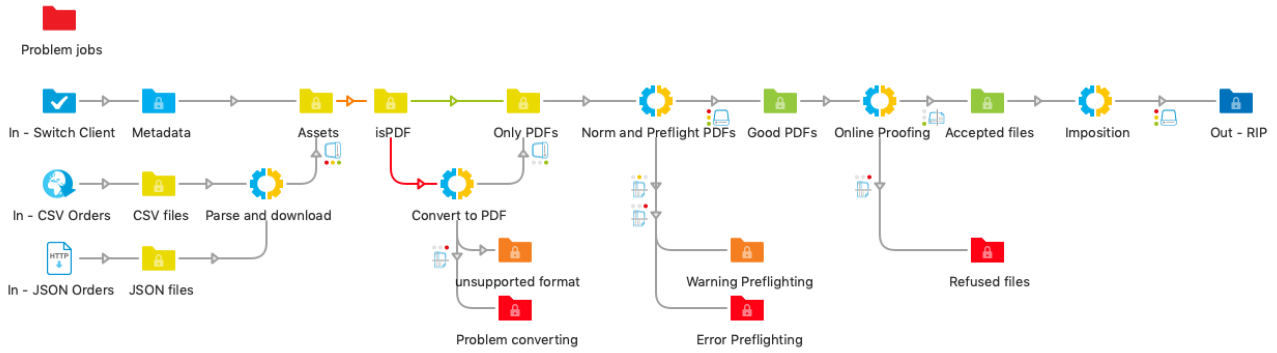
Amount of time (in minutes) after which an incoming job will fail if it has not been sent to the corresponding "Subflow - Receive" element.

The job will be retried every 15s.

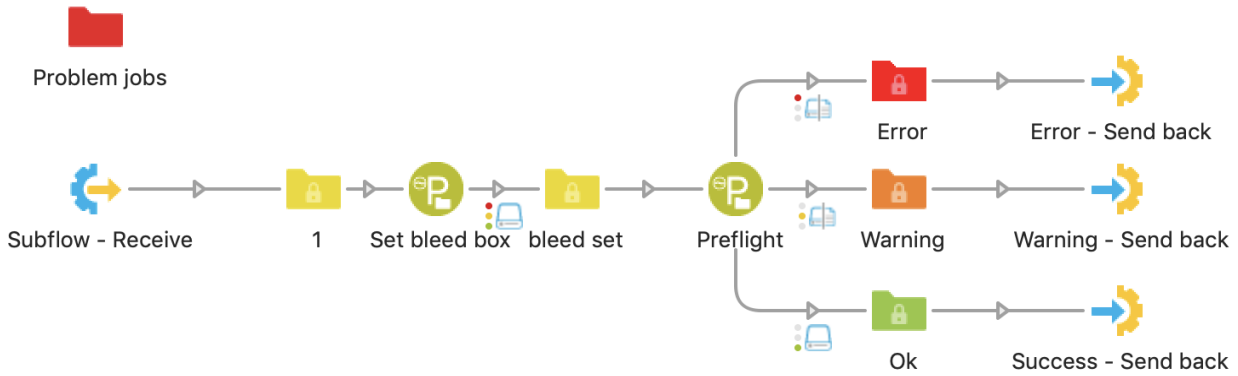
A value of 0 means this option is disabled.

Example flows

Main workflow



Norm and preflight subflow



Troubleshooting

- **Why doesn't my flow appear in the library from Subflow-Main?**

Only flows with a receive element are listed. (↔).

Flows that have been stopped for more than 7 days are removed from the list.

If the flow has been renamed, it must be activated at least once to appear on the list again.

- **Why does my job fail in the problem folder?**

Job can go in the fail folder if :

- The targeted flow does not exist or have been renamed
- The internal private data “**__subflow_callback__**” have been altered or is not present anymore.
- Switch does not have enough write/read right for the scriptData folder.
- The targeted flow is not on the subflow list (does not have any receive element, or stopped for more that 7 days)

If a timeout is set, the job will try every 15s for the timeout duration before going to the problem folder.

- **If the flow crash during a transfer, where are my files?**

Since subflow V4, jobs are no longer transferred through the HTTP protocol, but through a temporary folder generated on the fly.

The temporary folder can be found in the scriptData folder in the dataroot :

Exemple .\Enfocus Switch Dataroot\ScriptData\Subflow

By activating debugLog you can see the exact location of the files.

- **I have changed the name of my “main flow” or removed the “subflow-main” while jobs were on the “subflow”.**

Sadly the jobs will lose their way home.

You can manipulate the **EnfocusSwitchPrivateDataTag.subflow.callbackChain** private data key in order to restore the new flowName.

To be sure how this private data is constructed, send a new job in the subflow, then check the private data.

It should be constructed like this :

`<Flow>---<Subflow - Main ID>/<Flow>---<Subflow - Main ID>/...`

- **I want to create a recursive flow, but it could not be activated or I can't see it in the list.**

Recursive flows are possible, but not advised without a way to exit the recursion loop. “Main” element library only shows flow with “receive” elements that have been activated at least once.

Note on update:

Compatibility from V3 to V4 is granted. No file will be lost during the update.

Note on downgrade:

If for any reason you want to go back from V4 to V3, a file stuck in a subflow may not work. In order to make them work again, you need to setup two private data with appropriate value before the “sendBack” élément :

Private data	Default value
__subflow_listener__	HTTP
__subflow_protocol__	51080
__subflow_callback__	[Job.PrivateData:Key="EnfocusSwitchPrivate DataTag.subflow.callbackChain"]