



**Enfocus BV**

Kortrijksesteenweg 1095  
9051 Gent  
Belgium

+32 (0)9 216 98 01  
info@enfocuss.com

# Enfocus Switch 2021 Spring

## Release Notes

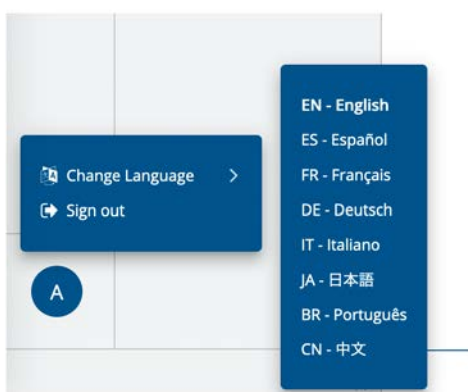
<b>LOCALIZATION</b> .....	<b>2</b>
SPANISH AND PORTUGUESE .....	2
<b>ELEMENTS</b> .....	<b>2</b>
SHOW ELEMENT TYPE .....	2
ARCHIVE ELEMENT: ENCRYPTION METHOD .....	3
EXECUTE COMMAND: STORE STDOUT,STDERR, AND EXIT CODE AS PRIVATE/METADATA .....	3
PROBLEM JOBS: RETRY ALERT MAILS .....	3
PROBLEM JOBS: DISABLE ALERT MAILS ON FLOW LEVEL .....	4
<b>SWITCH WEB PORTAL</b> .....	<b>4</b>
NEW! CUSTOM BRANDING .....	4
Redesign FLOW CONTROL .....	5
<b>SUBMIT POINTS AND CHECKPOINTS</b> .....	<b>6</b>
DEFINE FILETYPES ACCEPTED BY A SUBMIT POINT .....	6
NEW! MANAGE FILE LIST BEFORE UPLOADING .....	7
SHARING CHECKPOINT FILTERS + EXPORT AND IMPORT .....	8
Node.js <b>SCRIPTING</b> .....	<b>8</b>
CREATE NEW SCRIPT FOLDERS IN DESIGNER .....	8
LOGLEVEL.DEBUG .....	9
ADDITIONAL PRIVATE TAGS USED BY BUILT IN ELEMENTS .....	9
PDFDOCUMENT AND PDFPAGE CLASS .....	10
FLOWELEMENT CLASS - GET JOBS .....	11
<b>BUGFIXES</b> .....	<b>11</b>



## Localization

### Spanish and Portuguese

Spanish and Portuguese are added to the list of supported languages for Switch Designer, Switch Server, and the Switch Web Portal.



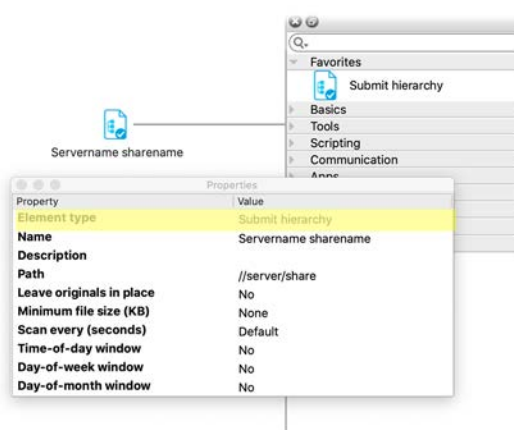
*List of all available languages in the Switch Web Portal.*

## Elements

### Show Element type

Improvements

A read-only property added to all elements used in a flow. The new property “Element type” displays the name of the element as it is shown in the Flow elements pane.



*Example of a Submit hierarchy element*

## Archive element: Encryption method

Improvements

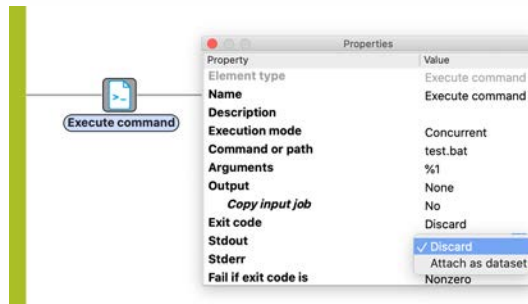


*Choose the Encryption Method after defining a password.*

## Execute Command: store output as private data/metadata

Improvements

Stdout and stderr are now optionally attached as a dataset, while the exit code can be stored as private data.

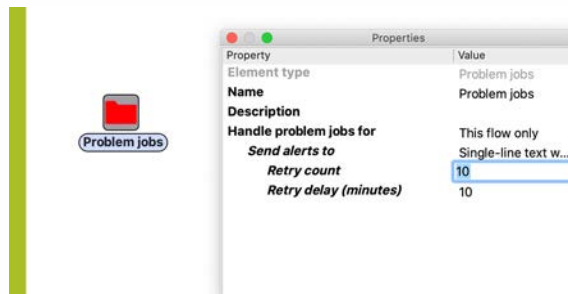


*Stdout example*

## Problem jobs: retry alert mails

Improvements

The new property is also available on server level in the user preferences.

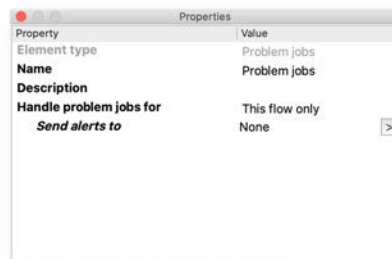


*Retry 10 times every 10 minutes*



## Problem jobs: disable alert mails on flow level Improvements

Disable problem job alert mails on flow level by adding a Problem jobs folder and selecting “None” for the “Send Alerts to” property.

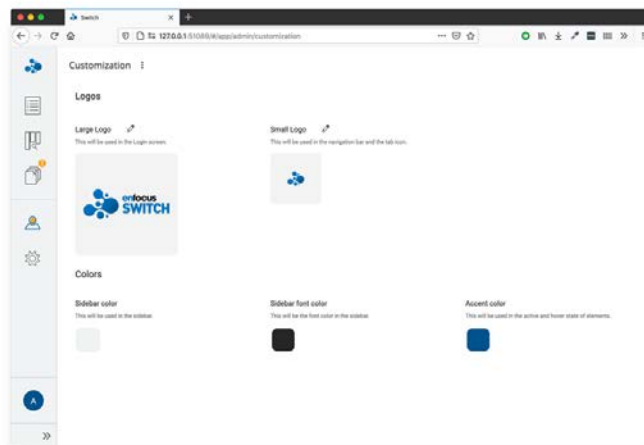


*This Problem jobs folder will not send any email alerts.*

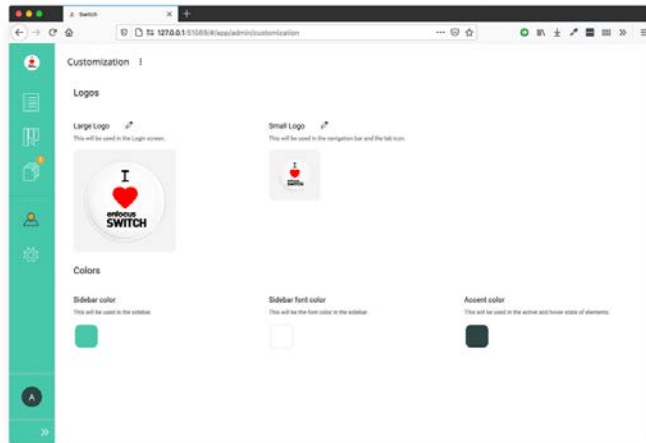
## Switch Web Portal

### Custom Branding NEW!

Administrators can configure the look of the Switch Web Portal by changing colors and logos.



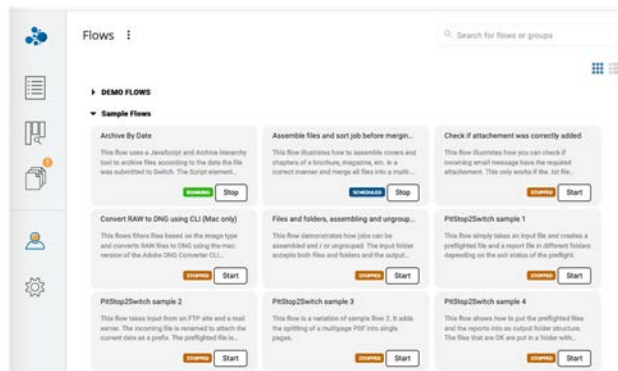
*The default configuration*



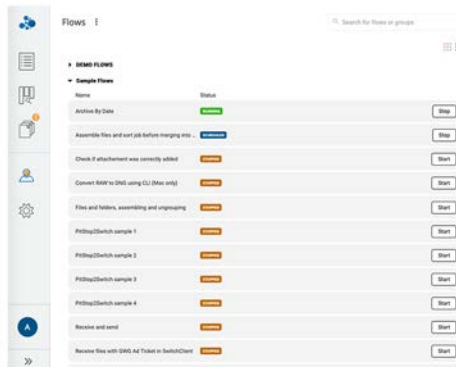
A custom branded Switch Web Portal

## Flow control Redesign

The flow control view in the Web Portal is redesigned to be quicker to use and offer a more consistent design between Web Portal views.



Card View including Flow description and labels for running/stopped/scheduled status.

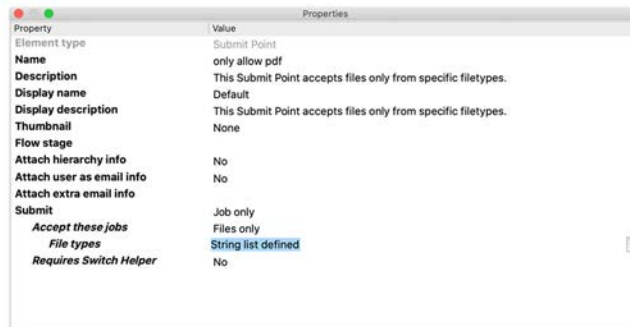


*A compact List View.*

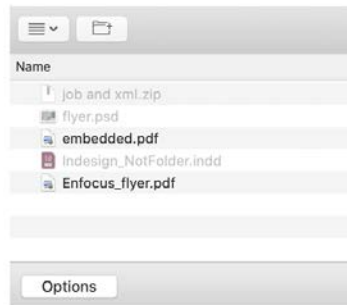
### Submit points and Checkpoints

Define file types accepted by a Submit point Improvement

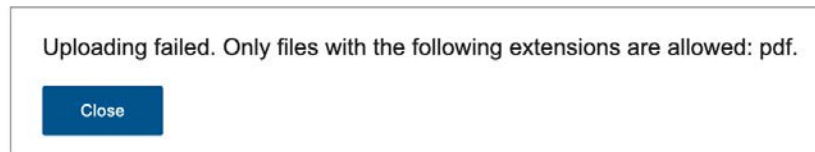
Submit points can now be configured to only accept folders and/or certain file types.



*Submit point configured to only accept certain file types.*



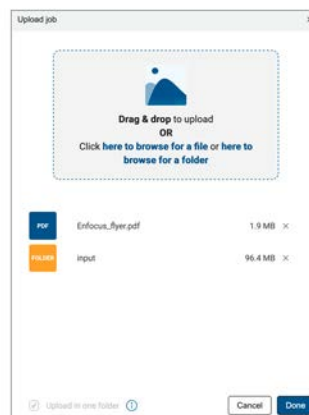
Only the accepted file types are enabled when using a file browse.



A warning is shown when dropping an unallowed file type on a Submit point

### Manage file list before uploading NEW!

You can now add and remove files (and folders) before opening the metadata dialog and starting the upload process.



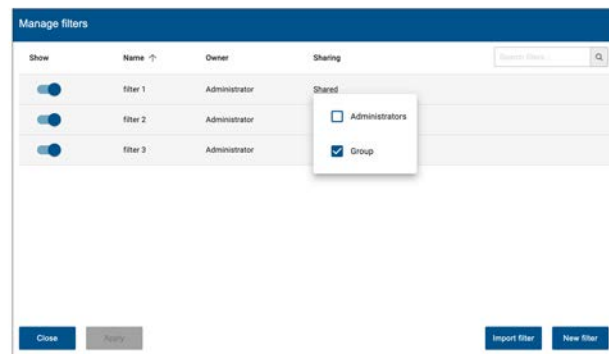
Manage the list before continuing to the metadata dialog.



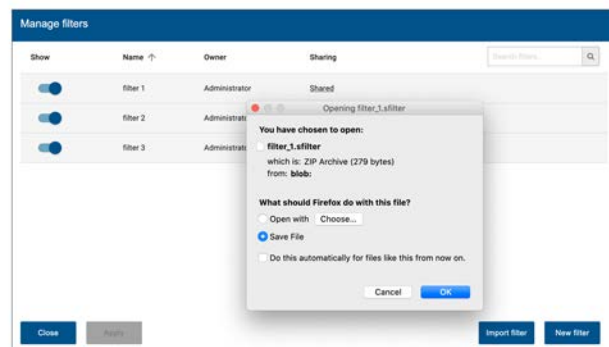
Share Checkpoint filters + export and import them

NEW!

Share custom filters for jobs with other users.



*A list of custom filters shared with a group of users.*



*Exporting and downloading a filter.*

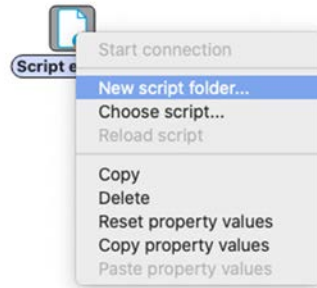
### Scripting

Create new script folders in Designer

Node.js

You can now create new script folders from within Switch Designer, without needing the SwitchScriptTool application. The feature is available from the contextual menu of a Script element on the canvas.



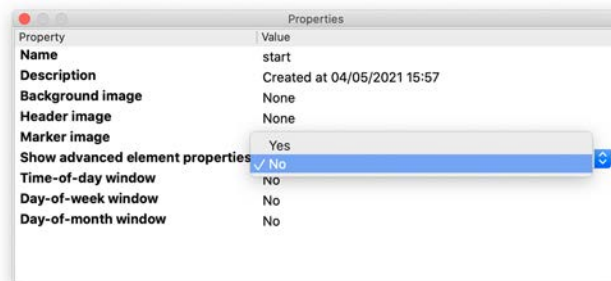


Create a new script folder using the contextual menu of Script elements

LogLevel.Debug

Node.js

Enable debug logging by first enabling advanced element properties on Flow level and afterwards debug logs on the element level containing the Node.js script.



First enable the advanced element properties on flow level

Additional private tags used by built-in elements

Node.js

Some private data tags are reserved for Switch built-in elements and can be read by Switch scripts. Except for `EnfocusSwitch.origin` these private data fields can be changed by scripts using `job.setPrivateData` calls.

*EnfocusSwitch.hierarchy(string[])*

An array with the location path segments in the hierarchy info associated with the job, or an empty array if there is no hierarchy info. The topmost path segment is stored at index 0.

*EnfocusSwitch.emailAddresses (string[])*

An array with the email addresses in the email info associated with the job, or an empty array if there are none.

*EnfocusSwitch.emailBody (string)*

The email body text in the email info associated with the job, or an empty string if there is none.



*EnfocusSwitch.userName (string)*

The short user name for the job, or an empty string if no user information has been associated with this job.

*EnfocusSwitch.userFullName (string)*

The full user name for the job, or an empty string if no user information has been associated with this job.

*EnfocusSwitch.userEmail (string)*

The user e-mail address for the job, or an empty string if no user information has been associated with this job.

*EnfocusSwitch.origin (string)*

An indication of the origin of the job before it was injected in the flow.

## PdfDocument and PdfPage Class Node.js

Retrieve PDF file and PDF page information.

*PdfDocument Methods examples:*

getNumberOfPages(): number;

Returns the number of pages in the PDF document.

getPDFVersion(): string;

Returns the version of the PDF file format (for example: "1.6").

getPDFXVersion(): string;

Returns the PDF/X version of the document, or an empty string if there is no PDF/X version.

And more...

*PdfPage Methods examples:*

getWidth(effective?: boolean): number;

Returns the width of the PDF page, in points. It is the same as getMediaBoxWidth.

getMediaBoxHeight(effective?: boolean): number;

Returns the height of the media box of the PDF page, in points. If effective is true, then the page rotation and scaling are taken

getCropBoxHeight(effective?: boolean): number;

Returns the height of the crop box of the PDF page, in points.

getBleedBoxWidth(effective?: boolean): number;

Returns the width of the bleed box of the PDF page, in points. If effective is true, then the page rotation and scaling are taken

And more...



```
if (numPages > 1) {
  // Get page boxes for the first page
  pdfPage = pdfDoc.getPage(1);
  abHeight = pdfPage.getArtBoxHeight(false);
  abWidth = pdfPage.getArtBoxWidth(false);
  bbHeight = pdfPage.getBleedBoxHeight(false);
  bbWidth = pdfPage.getBleedBoxWidth(false);
  cbHeight = pdfPage.getCropBoxHeight(false);
  cbWidth = pdfPage.getCropBoxWidth(false);
  mbHeight = pdfPage.getMediaBoxHeight(false);
  mbWidth = pdfPage.getMediaBoxWidth(false);
  tbHeight = pdfPage.getTrimBoxHeight(false);
  tbWidth = pdfPage.getTrimBoxWidth(false);
}

// Go over all the pages
for (let i = 2; i <= numPages; ++i) {
  pdfPage = pdfDoc.getPage(i);

  // Compare page boxes
  if (pdfPage.getArtBoxHeight(false) !== abHeight) ||
    (pdfPage.getBleedBoxHeight(false) !== bbHeight) ||
    (pdfPage.getCropBoxHeight(false) !== cbHeight) ||
    (pdfPage.getMediaBoxHeight(false) !== mbHeight) ||
    (pdfPage.getTrimBoxHeight(false) !== tbHeight) ||
    (pdfPage.getArtBoxWidth(false) !== abWidth) ||
    (pdfPage.getBleedBoxWidth(false) !== bbWidth) ||
    (pdfPage.getCropBoxWidth(false) !== cbWidth) ||
    (pdfPage.getMediaBoxWidth(false) !== mbWidth) ||
    (pdfPage.getTrimBoxWidth(false) !== tbWidth) {
    pageBoxesEqual = false;
    break;
  }
}
```

Code snippet using PdfPage methods

## FlowElement class - get jobs

Node.js

Returns a list of jobs waiting in the input folders of the flow element. The list includes all jobs that have "arrived" in the jobArrived entry point.

```
async function timerFired(s, flowElement) {
  await flowElement.setTimerInterval(30); // interval for 'holding' jobs
  const jobsInfo = await s.getGlobalData(Scope.FlowElement, 'jobsInfo');
  const currentTimestamp = new Date().getTime();
  let filteredIds = [];
  for (let id in jobsInfo) { // choose only jobs that have been staying in the flow
    element for at least 1 hour
    if ((jobsInfo[id] + (60 * 60 * 1000)) <= currentTimestamp && filteredIds.length
    <= 10000) {
      filteredIds.push(id);
      delete jobsInfo[id];
    }
  }
  if (filteredIds.length > 0) {
    const jobs = await flowElement.getJobs(filteredIds); // note: returns max 10000
    jobs
    for (let job of jobs) {
      await job.sendToSingle(); // move each job to next flow's point
    }
    await s.setGlobalData(Scope.FlowElement, 'jobsInfo', jobsInfo);
  }
}
```

Code snippet using getJobs to release multiple jobs at once

## BugFixes

ENFS-28525	Extra files removed from script folders (eg: versioning/git related files)
ENFS-28260	Temp folder not being cleaned up
ENFS-28121	Remote Designer crashing when connecting from Mac to Windows
ENFS-27740	Node.js scripts: properties with dynamic values are evaluated for any entry point



ENFS-27718	Heavily loaded Archive hierarchy element blocks other Archive hierarchies
/ENFS-27717	Improve Switch Helper open file error message
ENFS-27670	Leave originals sometimes ignored
ENFS-27610	Input field size checkpoint in dependent fields not adjusted.