

# Design document

## **Telefonica/O2** **File Format Validation Custom AR step**

Version 1.00 of January 20, 2022

Status: Preliminary Draft

---

# Summary

---

1.	Overview	3
1.1.	<b>Functional Overview</b>	<b>3</b>
1.2.	<b>Architectural Overview</b>	<b>3</b>
2.	Features and Implementation	4
2.1.	<b>File Format Validation Custom AR Step</b>	<b>4</b>
2.2.	<b>Step Settings:</b>	<b>4</b>
3.	Notes, Assumptions and Limitations	6
4.	Example flow	7
4.1.	<b>Validating XML files before sending to target destination</b>	<b>7</b>
4.2.	<b>Flow Configuration</b>	<b>7</b>

## 1. Overview

---

This chapter briefly describes the features of the solution package.

### 1.1. Functional Overview

The **File Format Validation** Custom Advanced Routing (AR) step will perform file format validation, depending on the specified file type. The files, which are considered invalid, will be renamed with extension “.refused”. The selected file type, result of the validation (Accepted | Refused), error message (in case of an invalid file) and the specified quarantine folder (in case of an invalid file) will be exposed as flow attributes to the file. All processed files, both valid and invalid, will be returned as a result from the step. This result can further be used as input to other OOB AR steps.

The implementation will have a couple of aspects:

- The step will be based on Custom AR step **SPI v1.1**.
- The custom step will be displayed as "**File Format Validation**" in ST AR Admin UI.
- The custom step will be available for configuration in *ST Admin UI* → *Account* → *Routes* page.
- The custom step will be available for configuration by *ST REST API*.

### 1.2. Architectural Overview

- Operating System: *all OS supported by ST*
- Base Product version: *SecureTransport 5.5*
- Base Custom AR step SPI version: *1.1*

## 2. Features and Implementation

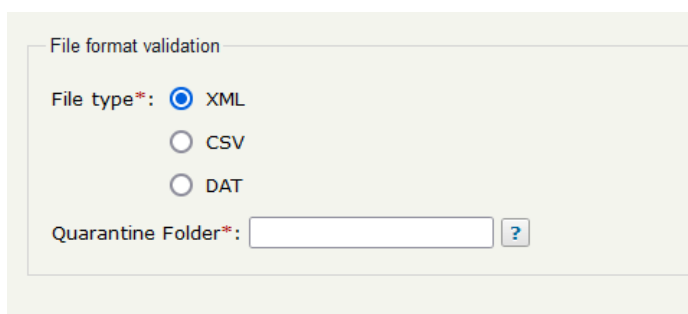
This chapter describes the required features of the system, the intended use scenarios for these features and their implementation details.

### 2.1. File Format Validation Custom AR Step

The **File Format Validation** custom step will perform file format validation, depending on the specified file format. The step will rename the invalid files, adding a “.refused” extension to the name of each invalid file. All processed files, both valid and invalid, will be returned as a result from the step. This result can further be used as input to other OOB AR steps. The selected file type, result of the validation (*Accepted* | *Refused*), error message (in case of an invalid file) and the specified quarantine folder (in case of an invalid file) will be exposed as flow attributes to the file to be used in subsequent OOB AR steps, as well as the Sentinel reporting.

### 2.2. Step Settings:

- **File type** – required field – specifies the type of the file against which it will be validated. Valid values are: *xml*, *csv* and *dat*. The property value will be set as a file flow attribute.
- **Quarantine Folder** – required field – The folder where the invalid files should be moved. The property value will be set as a file flow attribute.
- **Additional settings, depending on the file type:**
  - **XML** file type: no additional settings.



- **CSV** file type:
  - **Separator** - required field for **.csv** file type – the chosen separator for .csv file type.
  - **Element Number** – for **.csv** file type – Number of expected elements in the .csv file. Each record of the file must contain the specified number of elements. If there is no provided value, the element number will be the number of elements of the first record of the file.

File format validation

File type\*: ☐ XML  
☒ CSV  
☐ DAT

Quarantine Folder\*:  ?

CSV settings

Separator\*:  ?  
Element number:  ?

- **DAT** file type:
  - **Fixed Length** - for **.dat** file type – Fixed length of the file records. Each record of the .dat file must contain the specified number of symbols. If there is no provided value, validation against this field will not be performed.
  - **Record Header** - for **.dat** file type – The specified value will be checked as a record header for the file. If there is no provided value, validation against this field will not be performed.
  - **Forbidden Characters** - for **.dat** file type – Forbidden characters in the file. By default, the comma char ‘,’ is considered forbidden. If no value is provided, the file will be scanned for comma ‘,’; otherwise – it will be scanned for ‘,’ along with the other specified symbols.

File format validation

File type\*: ☐ XML  
☐ CSV  
☒ DAT

Quarantine Folder\*:  ?

DAT settings

Fixed length:  ?  
Record Header:  ?  
Forbidden characters: ,  ?

### 3. Notes, Assumptions and Limitations

---

- The step **DOES NOT MOVE** the invalid files. The step renames the invalid files, adding a “.refused” extension to the name of each invalid file. All processed files, both valid and invalid, will be returned as a result from the step to give the opportunity for further transformation/routing of them via other OOB AR steps.
- Quarantine folder is used **ONLY** to be set as a flow attribute to the file. The quarantine folder flow attribute can be used in a following step with the expression:  
`${flow.attributes['quarantineFolder']}`.
- File type, status of the validation (Accepted | Refused), error message (in case of an invalid file) and quarantine folder (in case of an invalid file) will be set as file flow attributes. After validation of the file if it is considered as valid, then no flow attributes will be populated for the error message and the quarantine folder. The attributes then can be **reported to Sentinel using the additional Sentinel reporting capabilities of ST via Sentinel Attribute Mapping**. The flow attributes must be mapped to an existing Sentinel column and will be reported in **AVAILABLE** state in the existing **XFBTransfer** tracking object.
- The custom step fields will support ST expression language.
- All custom step properties will have a tooltip with description and additional information.
- **Proposed solution considers that after the validation the valid files will be sent to a chosen location via the OOB AR step, while the invalid files will be sent to another specified location via OOB AR step. The subscription folder of the custom AR step cannot be the desired destination folder for the valid and the invalid files.**
- No migration utilities are covered under this implementation.



- Configure the **xmlFilesValidation** route to be executed only for files with “.xml” extension. This could be done using condition with the following expression:  
`${transfer.target.matches('*.xml')} ? 1 : 0}`
- Create 3 steps for the route – **File Format Validation**, which will validate the XML files and will rename the invalid files adding the “.refused” extension to their file names, **Publish To Account** – to send the invalid files to a quarantine folder, and another **Publish To Account** to send the valid files to a desired location.

User Account : source1

**Update Route** Save Cancel

Route Name: \* xmlFilesValidation

Description:

Condition: ☐ Always ☒ Expression Language  
`${transfer.target.matches('*.xml')} ? 1 : 0` ?

**Steps** -- Select Step -- Add Step

☒ Enable ☒ Disable ☒ Reorder ☒ Delete

Step Type
<input checked="" type="checkbox"/> <a href="#">File Format Validation</a>
<input checked="" type="checkbox"/> <a href="#">Publish To Account</a>
<input checked="" type="checkbox"/> <a href="#">Publish To Account</a>

*Steps will be applied in the order they are listed here.*

**Notifications**

☐ Notify following e-mails on route failure:  ?  
Mail Template: None ?

☐ Notify following e-mails on route success:  ?  
Mail Template: None ?

☐ Notify following e-mails on route triggering:  ?  
Mail Template: None ?

\* Indicates required field  
Enter value or expression

Save Cancel



- Configure the **File Format Validation** step. Select the 'XML' file type and enter the path to the quarantine folder.



Input Files

☒ Process only result from preceding step ?

Name Filter\*: ☒ Process all files  
☐ Process files based on a file name pattern

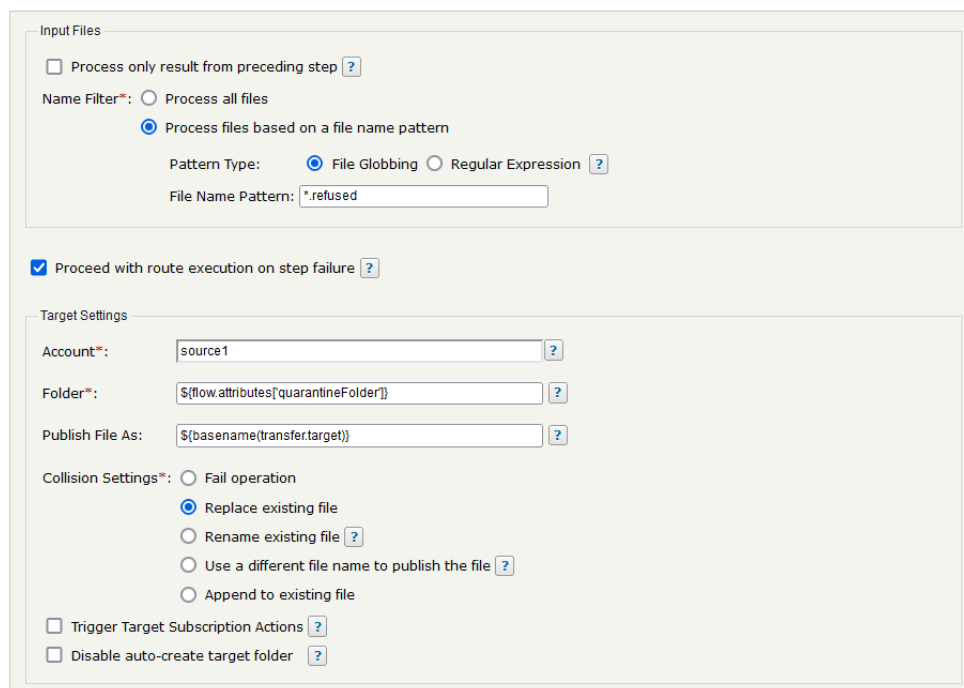
☐ Proceed with route execution on step failure ?

File format validation

File type\*: ☒ XML  
☐ CSV  
☐ DAT

Quarantine Folder\*: /invalid/xml ?

- Configure the first **Publish To Account** step, which will send the invalid files to a quarantine folder. The invalid files are renamed from the **File Format Validation** step and have the ".refused" extension. To route only the invalid files set the File Name Pattern to: \*.refused . Use the expression for the quarantine folder: `${flow.attributes['quarantineFolder']}`, to configure the destination folder. To send the invalid files with their original file name, set the Publish File As to: `${basename(transfer.target)}`.



Input Files

☐ Process only result from preceding step ?

Name Filter\*: ☐ Process all files  
☒ Process files based on a file name pattern

Pattern Type: ☒ File Globbing ☐ Regular Expression ?

File Name Pattern: \*.refused

☒ Proceed with route execution on step failure ?

Target Settings

Account\*: source1 ?

Folder\*: \${flow.attributes['quarantineFolder']} ?

Publish File As: \${basename(transfer.target)} ?

Collision Settings\*: ☐ Fail operation  
☒ Replace existing file  
☐ Rename existing file ?  
☐ Use a different file name to publish the file ?  
☐ Append to existing file

☐ Trigger Target Subscription Actions ?

☐ Disable auto-create target folder ?

- Configure second **Publish To Account** step, which will send to target destination only the valid files. The valid files are NOT renamed from the **File Format Validation** step, so their file name does not end with the “.refused” extension. To route only the valid files set the Pattern Type to Regular Expression and the File Name Pattern to: `.*\.(?!refused)[^.]+`

Input Files

☐ Process only result from preceding step [?](#)

Name Filter\*: 
☐ Process all files
☒ Process files based on a file name pattern

Pattern Type: 
☐ File Globbing 
☒ Regular Expression [?](#)

File Name Pattern:

☒ Proceed with route execution on step failure [?](#)

Target Settings

Account\*:  [?](#)

Folder\*:  [?](#)

Publish File As:  [?](#)

Collision Settings\*: 
☐ Fail operation
☒ Replace existing file
☐ Rename existing file [?](#)
☐ Use a different file name to publish the file [?](#)
☐ Append to existing file

☐ Trigger Target Subscription Actions [?](#)

☐ Disable auto-create target folder [?](#)