Plugin for importing processes via Hotfolder

Introduction

This documentation describes the installation, configuration and use of the plugin.

Details	
Identifier	plugin_intranda_administration_hotfolder_nli
Source code	https://github.com/intranda/plugin_intranda_administration_hotfol
Licence	GPL 2.0 or newer
Compatibility	Goobi workflow 2021.09
Documentation date	23.10.2021

Installation

The program consists of the files:

```
plugin_intranda_administration_hotfolder_nli.jar
plugin_intranda_administration_hotfolder_nli-GUI.jar
plugin_intranda_administration_hotfolder_nli.xml
```

The file `plugin_intranda_administration_hotfolder_nli.jar` contains the program logic, and should be copied to this path: `/opt/digiverso/goobi/plugins/administration/`.

The file `plugin_intranda_administration_hotfolder_nli-GUI.jar` contains the graphical interface, and should be copied to this path: `/opt/digiverso/goobi/plugins/GUI/`.

The file `plugin_intranda_administration_hotfolder_nli.xml` is the config file, and should be copied to the folder `/opt/digiverso/goobi/config/`.

Configuration

The configuration is done via the configuration file `plugin_intranda_administration_hotfolder_nli.xml` and can be adapted during operation. It is structured as follows:

```xml
<config_plugin>

    <!-- hotfolder -->
    <hotfolderPath>/opt/digiverso/import/hotfolder</hotfolderPath>

    <config>
        <!-- Use this config for the following workflow template: -->
        <template>NLI-template</template>

        <!-- publication type to create -->
        <publicationType>Manuscript</publicationType>
```

```
<!-- which digital collection to use -->
<collection>NLI</collection>

<!-- define if a catalogue shall get requested to import metadata -->
<useOpac>true</useOpac>

<!-- which catalogue to use (as default) -->
<opacName>Alma</opacName>

<!-- which catalogue to use per record; if missing the default will be used -->
<opacHeader>UserDefinedB</opacHeader>
<searchField>12</searchField>

<!-- define in which row the header is written, usually 1 -->
<rowHeader>1</rowHeader>

<!-- define in which row the data starts, usually 2 -->
<rowDataStart>2</rowDataStart>

<!-- define in which row the data ends, usually 20000 -->
<rowDataEnd>20000</rowDataEnd>

<!-- define which column is the one to use for catalogue requests -->
<identifierHeaderName>Identifier</identifierHeaderName>

<!-- define which column is the one to use for naming the process -->
<processHeaderName>Process title</processHeaderName>

<!-- define which column is the one to use for naming the image files -->
<imagesHeaderName>dcterms:IE_Title</imagesHeaderName>

<!-- Overwrite any existing processes -->
<replaceExistingProcesses>true</replaceExistingProcesses>

<!-- define here which columns shall be mapped to which ugh metadata
    ugh: name of the metadata to use. if it is empty or missing, no metadata is gen
    headerName: title inside of the header column
    property: name of the process property. if it is empty or missing, no process p
    normdataHeaderName: title of the header column to use for a gnd authority ident
    docType: define if the metadata should be added to the anchor or child element.
    record is no multivolume. Default is 'child', valid values are 'child' and 'anc
-->
<metadata
    ugh="CatalogIDDigital"
    headerName="Identifier"
```

```
            mandatory="true" />
        <metadata
            ugh="CatalogIdentifier"
            headerName="UserDefinedB"
            mandatory="true" />
        <metadata
            ugh="SubjectTopic"
            headerName="UserDefinedA"
            mandatory="true" />
        <metadata
            headerName="UserDefinedC"
            mandatory="true" />
    </config>
</config_plugin>
```

Value	Description
`hotfolderPath`	The path to the Hotfolder.
`template`	Name of a workflow template. If a subfolder of that name exists with a .XSLX file, it will be imported.
`collection`	Name of the Collection which each new process should belong to.
`useOpac`	If true, use the Identifier of each entry to get metadata from the specified opac.
`opacName`	Name of opac to use by default.
`opacHeader`	If the name of a different opac is under this header in the Excel file, then use that opac instead.
`searchField`	Field to search in the specified opac.
`rowHeader`	The entry rowHeader determines in which row the plugin will look for the name to display with this column.
`rowDataStart`	The entry rowDataStart tells the plugin which row is the first containing the content to parse.
`rowDataEnd`	The entry rowDataEnd tells the plugin which row is the last containing content. Rows before this may be empty.
`identifierHeaderName`	Title of the column in which the plugin will look for the Identification for the entry.

Value	Description
`processHeaderName`	Title of the column specifying the second part of the name of the process to be created.
`imagesHeaderName`	If this has an entry for a given row, then the images will be renamed after the text in this cell.
`replaceExistingProcesses`	If true, replace any existing process with the same name. If false, do not create a new process if there already is one of that name.

Configuration of the individual metadata fields

For each individual metadata, we specify how it is to be imported. Each field corresponds to the contents of a cell in an Excel column. The following values are possible.

Value	Description
`headerName`	This tells the plugin which column this entry refers to.
`ugh`	This tells the plugin which metadatum the column should be written to (e.g. CatalogIDDigital, Country or Institution).
`mandatory`	If this is true, then any row of the Excel file which does not have a value in this column will not be imported.

Operation of the plugin

The hotfolder to be observed is defined by the entry `hotfolderPath`. For each subfolder, if the name of the subfolder coincides with the `template` entry of any subconfiguration, then look in this folder. In the example above, look in the folder

`/opt/digiverso/import/hotfolder/NLI-template/`

For each subfolder `sub` inside this folder, look inside `sub`. In each such folder, the plugin looks for an Excel `.XLSX` file. If it finds one, then each row of the excel file is read, and if it is appropriately configured, then a new process is created in Goobi. The name of the process is put together with the name of the intermediate folder `sub`, and the `processHeaderName` column of the corresponding row.

A new process is created as follows. Consider one row of the Excel file, for example with `Identifier` entry “9001234” and `processHeaderName` entry `100-2`. First the plugin checks that the row contains values for every mandatory column. Next, it looks for a subfolder of `sub` with the name `100-2`. If this exists, and it contains images, and it was not modified in the last half hour, then a new process will be created. This is named “sub_100-2” after the intermediate folder and the process name entry. Metadata will be imported from the specified opac using the identifier “9001234”. All metadata read from cells of the row are also written, and if they conflict with data from the opac, then the opac data is overwritten.

All images in the subfolder `sub/100-2/` will be copied to the new process, and renamed. If the row has an entry corresponding to `imagesHeaderName`, eg “Gdl-1-12”, then the images will be sorted by name, and then renamed as

Gdl-1-12_0001.tif , Gdl-1-12_0002.tif , . . .

If the row does not have an entry corresponding to `imagesHeaderName`, then the images are renamed after the process name, eg in this example

sub_100-2_0001.tif , sub_100-2_0002.tif , . . .

Finally, the folder `sub/100-2/` is deleted.

Graphical interface

The graphical interface can be found in the goobi workflow “Administration” menu under the entry “NLI Hotfolder Import”. Here the running of the plugin can be paused by clicking on “Pause work”, and resumed again by clicking on “Resume work”. It also shows the running time of the plugin.