Rosetta CSV creation tool instructions

Scope of functions

The main function of this tool is to convert given simple CSV files into Rosetta CSVs, ready for deposit. The tool is specifically useful for digitization projects where there is a *unique ID* per book (or any other intellectual entity - IE), which is used as base name (prefix) for all created files of this IE.

The tool
- searches folders for related files
- adds the required lines for representations and files
- adds columns on representation and file level
- extracts file label from file name via regular expression
- retains the order of IEs from the source CSV and sorts the files by file name
- offers specific functions depending on OS
  Windows: create a ZIP file of the stream file folders for upload
  UNIX: add full path of files to CSV, so that the files don't need to be copied to SIP directories but can be deposited from their original location
- has some flexibility because of configuration options
- writes a log file (with debug information, when configured)

Prerequisites

The expected source CSV has just one line per IE, which contains metadata fields on IE level. In addition, the first column must contain the base name (prefix) of related files.

Example:

Filename for matching	Object Type	Title (DC)	Creator (DC)	Title (DC)	IE Entity Type
	SIP	SIP title			
proj202107_0001_jproj202107_0001-addon_	IE		Creator of Title 1	Title 1	Book
proj202107_0002_	IE		Creator of Title 2	Title 2, with comma	Book
proj202107_0015_	IE		Creator of Title 15	Title 15, without Modified Master representation	Book
proj202107_0004_	IE		Creator of Title 4	Title 4, with comma	Book

All requirements are detailed in the configuration XML.

Configuration and Usage

The 'conf' folder exists next to the executable 'createRosettaCSV.jar' and contains the configuration file 'CreateRosettaCSV.xml'. All options are explained in the file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<properties>

<!--
IMPORTANT
Requirements for the provided CSV:
  - separator is comma (,)
  - first column must have header 'Filename for matching' (can contain multiple base
    names separated by pipe |)
  - first line contains header
  - header values must not contain comma
  - there can be only one line for object type SIP
  - COLLECTION lines are currently not supported
  - there is exact one line per IE
  - there are no other lines than for object type SIP or IE
  - all columns on representation and file level will be added by the tool and MUST NOT
    already exist in the original CSV:
    Preservation Type, File Original Path, File Original Name, File Label
    'Preservation Type' will be the SAME as the folder name that contains the streams of
    this representation and are configured in <repfolder>
    (e.g. <repfolder>PRESERVATION_MASTER,MODIFIED_MASTER,DERIVATIVE_COPY</repfolder>)
    'File Original Path'
       Unix: depends on <addfullpath>
       Windows: the representation folder name, optionally preceded by the path
                configured in <nfspathtostreams>
    'File Original Name' will be the file name
    'File Label' can be extracted from file name via regex

Output:
on Windows PC: converted CSV file (and optional ZIP file containing stream files)
on Unix server: SIP directory with expected structure and converted CSV file
 -->

<section name="source">
  <!-- <sourcerootcsv>:
  - folder containing the CSV file(s) for transformation
  - path can be absolute or relative (relative to the JAR file)
  -->
  <sourcerootcsv>CSV_source</sourcerootcsv>

  <!-- <sourcerootfiles>:
  - folder containing subfolders for each type of representation
  - path can be absolute or relative (relative to the JAR file)
  -->
  <sourcerootfiles>files_source</sourcerootfiles>

  <!-- <repfolder>:
  - subfolders containing file streams
  - the subfolder name also defines the 'Preservation Type' of the representation
  - multiple types are separated by comma
  -->
  <repfolder>PRESERVATION_MASTER,MODIFIED_MASTER,DERIVATIVE_COPY</repfolder>
```

```
  <!-- <labelregex>:
  - regular expression that will be used to extract 'File Label' from 'File Original
Name'
  - if missing or empty, 'File Original Name' will be copied to 'File Label'
  Example:
    file name: xxxx_yyyy_label.jpg
    labelregex: ^.*?_.*?_(.*?)\..*$
    results in: label
  - the regex must match the complete filename
  - the value in brackets will be used as label
  - only one group, i.e. pair of brackets, is supported
  -->
  <labelregex>^.*_(.*?)\..*$</labelregex>
</section>

<section name="general">
  <!-- <addfullpath>:
  - only relevant when running the tool on Unix server (not locally in Windows)
  - true: full path information for source files is written into 'File Original Path'
    - NOTE: stream files can remain in their original location during deposit
  - false: relative path for {representation folder} + '/', e.g. 'PRESERVATION_MASTER/'
    is written into 'File Original Path'
    - NOTE: stream files must be copied/moved into SIP directory 'content/streams/...'
  -->
  <addfullpath>false</addfullpath>

  <!-- <zipstreamfolder>:
  - let the tool create a ZIP file of the stream file folders for upload (true / false)
  - only active on Windows PC
  -->
  <zipstreamfolder>true</zipstreamfolder>

  <!-- <debug>: add debug information to the log (true / false) -->
  <debug>false</debug>
</section>

<section name="target">
  <!-- <targetrootcsv>:
  - folder containing the CSV file ready for deposit
  - path can be absolute or relative (relative to the JAR file)
  - if <zipstreamfolder> is 'true', this folder will also contain the zipped stream
    folders
  -->
  <targetrootcsv>ready_for_deposit</targetrootcsv>

  <!-- <nfspathtostreams>:
  - absolute path to the NFS directory on the Unix server, containing the
    representation folders and stream files
  - this path is added as prefix to the 'File Original Path' column
  - only active when running the tool in Windows and parameter 'zipstreamfolder' is set
    to 'false'
  - NOTE: Representation folders with stream files must be uploaded into this location
    on the server
  -->
  <nfspathtostreams>/local_deposit_storage/all_streams_directory/</nfspathtostreams>
</section>

</properties>
```