Instructions for preparing and converting spreadsheet outlines to XML for upload to www.tbrc.org

File name formats:

File name format is important and must be adhered to since the conversion software makes use of parts of the name.

The CSV files are comma or tab-delimited.

The format for names of spreadsheet files and the corresponding CSV files is:

TBRC-Outline-Wxxxxxxx-b
TBRC-Outline-Wyyyyyyy-t
TBRC-Outline-Wzzzzzzz-c

The converter requires these conventions to be adhered to.

When we process a western *book* formatted Work we'll append "**-b**" (for book) to the end of the file name for the spreadsheet (and consequently for the tab delimited csv file as well). For a book the pagination is starts from the beginning of the

When a *per Text* pecha is processed the filename will have "**-t**" (for Text) appended to name.

When a *continuous* pecha is processed the filename will have "**-c**" (for continuous) appended to the name.

In the case of **b** or **t** type files there will be 8 fields as currently used.

For the **c** type files there will be two additional fields which will be folioStart and folioEnd (written in arabic numerals).

The types: b, t, c; control how the **<o:description/>** is formatted for the resulting Outline. When the converter processes a **b** type file it does not add the **xx ff.** to the **<o:description type='location'/>**; otherwise, for types **t** and **c** the folio information will be generated.

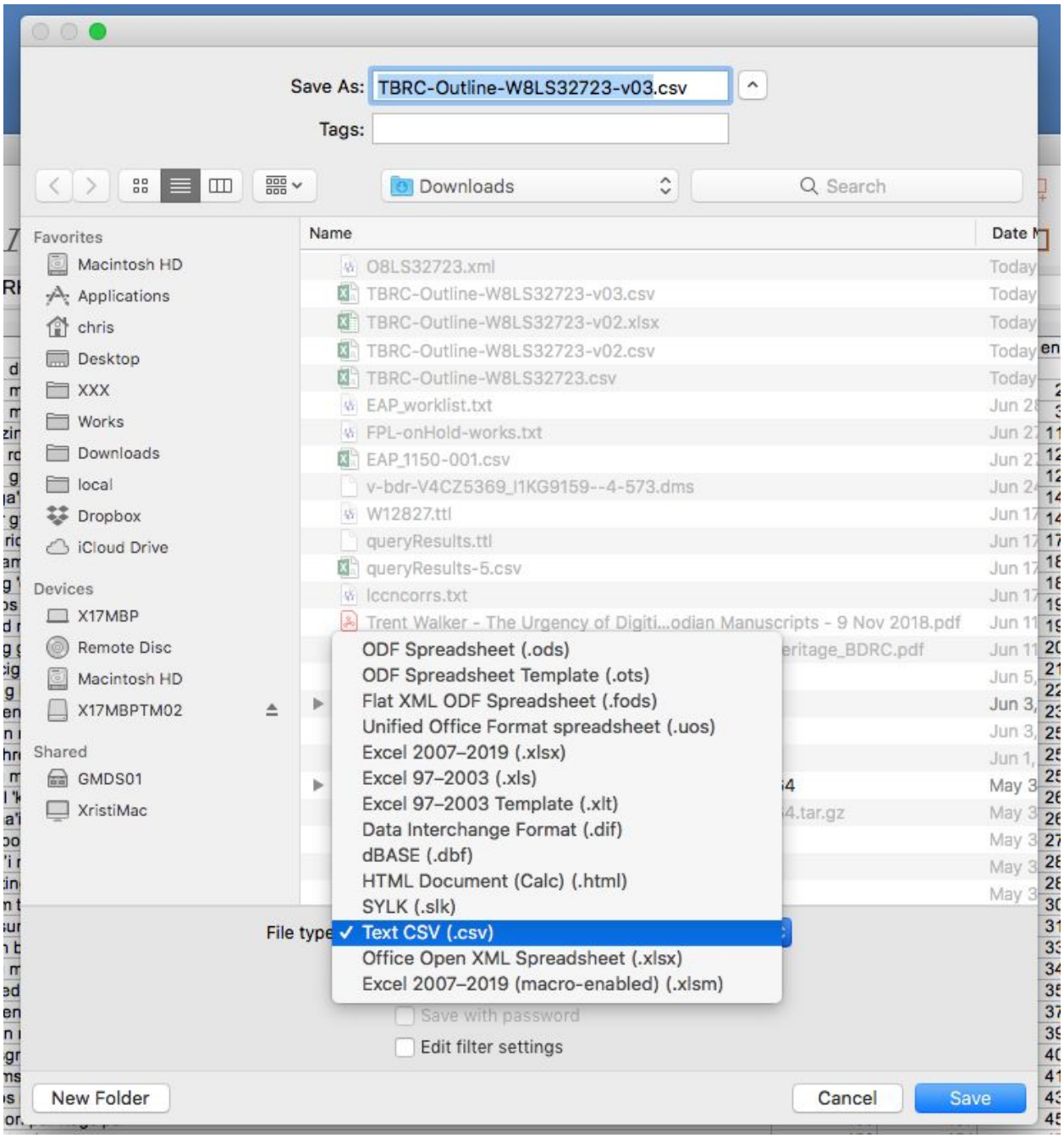
Since version 1.7.0 there is support for additional information in the csv. The ***extended*** mode is only applicable to the per text (default) mode. This is signalled by running the converter with the "`-extended`" flag. In this usage there are 3 additional columns:

1. Authors - a comma separated list of Person rids
2. Subjects - a comma separated list of rids of Topics, Places, Persons or Works that are subjects of this text
3. Note - a note to be included in the text node

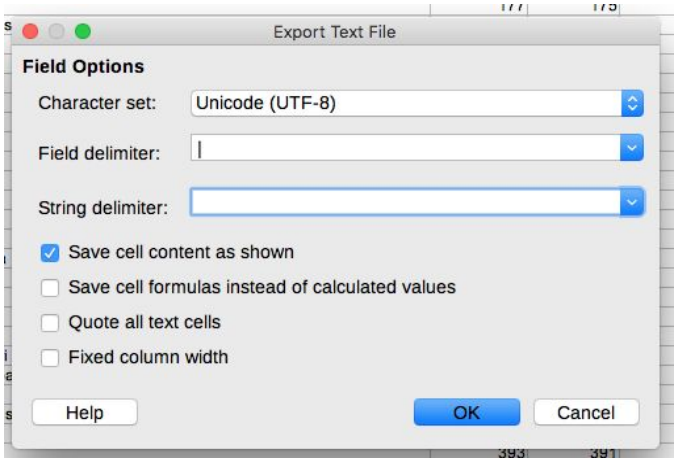
NB, the Work RID should only appear on the first row (after headings) and the volume number only on the first row of each volume; otherwise, a malformed Outline will be produced.

Preparing spreadsheets for conversion:

Open spreadsheets in an appropriate program, usually Libre Office or OpenOffice and save as tab or comma separated UTF-8 csv. For example, in Libre Office save via the **"Save As ..."** >> **"Text CSV ...":**



The "Field delimiter" that is used is the vertical bar character, **|**, and the "Text delimiter" is erased so that there is none. Most important is that the Character set be **UTF-8**:



The key is to save the content properly as **UTF-8** csv format for the conversion program.

The spreadsheets may also be xls, xlsx, and so on. No spaces and the names need to correctly use zeros not upper letter O in the Work number.

Running the converter:

The latest version of the converter is csv2outline-1.6.0.jar which is a standalone runnable jar. There is a shell script in the same folder as this file and the jar file:

```
csv2outline.sh
```

The script requires four parameters: 1) the input folder path of csv files to be converted, 2) the output folder path where the; 3) the db user name; and 4) the db password. Assuming that the terminal is in the folder containing the script and jar and the input and output folders are also in the same folder then the script can be run like:

```
./csv2outline.sh IN OUT CodeFerret "
```

The jar can be run without the script with the parameter `-help`:

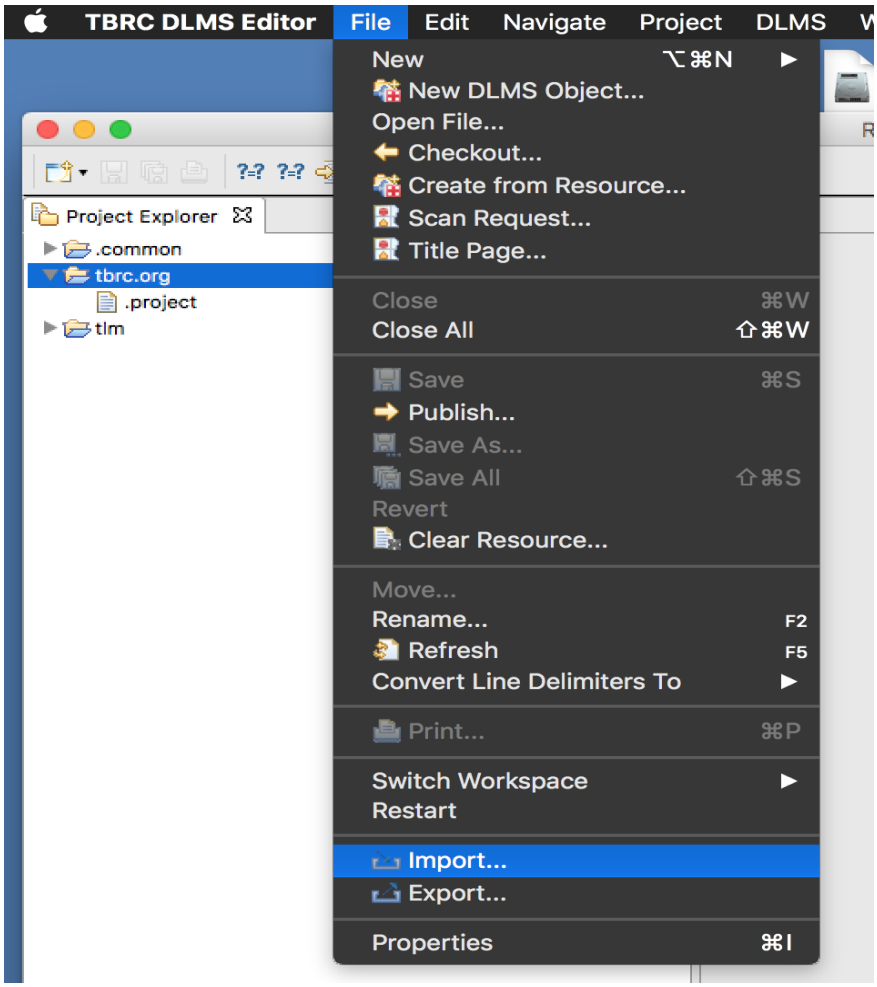
```
$ java -jar convert-2-outline-xml-1.6.0.jar -help
java -jar csv2outline.jar (-doc <pathname> | -docdir <pathname>) -outdir <pathname> -who <name> -title <title>

-help - print this message and exits
-version - prints the version and exits
-doc <pathname> - path to dkar chag in tab delimited csv
-docdir <pathname> - path to directory of dkar chag in tab delimited csv
-outdir <pathname> - path to base directory in which Outline XML will be written - no trailing slash '/'
-who <unicodename> - name of the person creating the dkar chag. Defaults to anon
-title <work title> - title of the work that the outline is for
-type <outline type> - outline type name. Defaults to 'subjectCollection'
-folio <book|text|cont> - form of folio information. Defaults to text
-extended - indicates an extended csv with up to 4 additional columns
-verbose - prints basic processing information
-debug - prints diagnostic information useful in debugging format problems
-trace - prints each token
```

Convert2OutlineXML version: 1.6.0

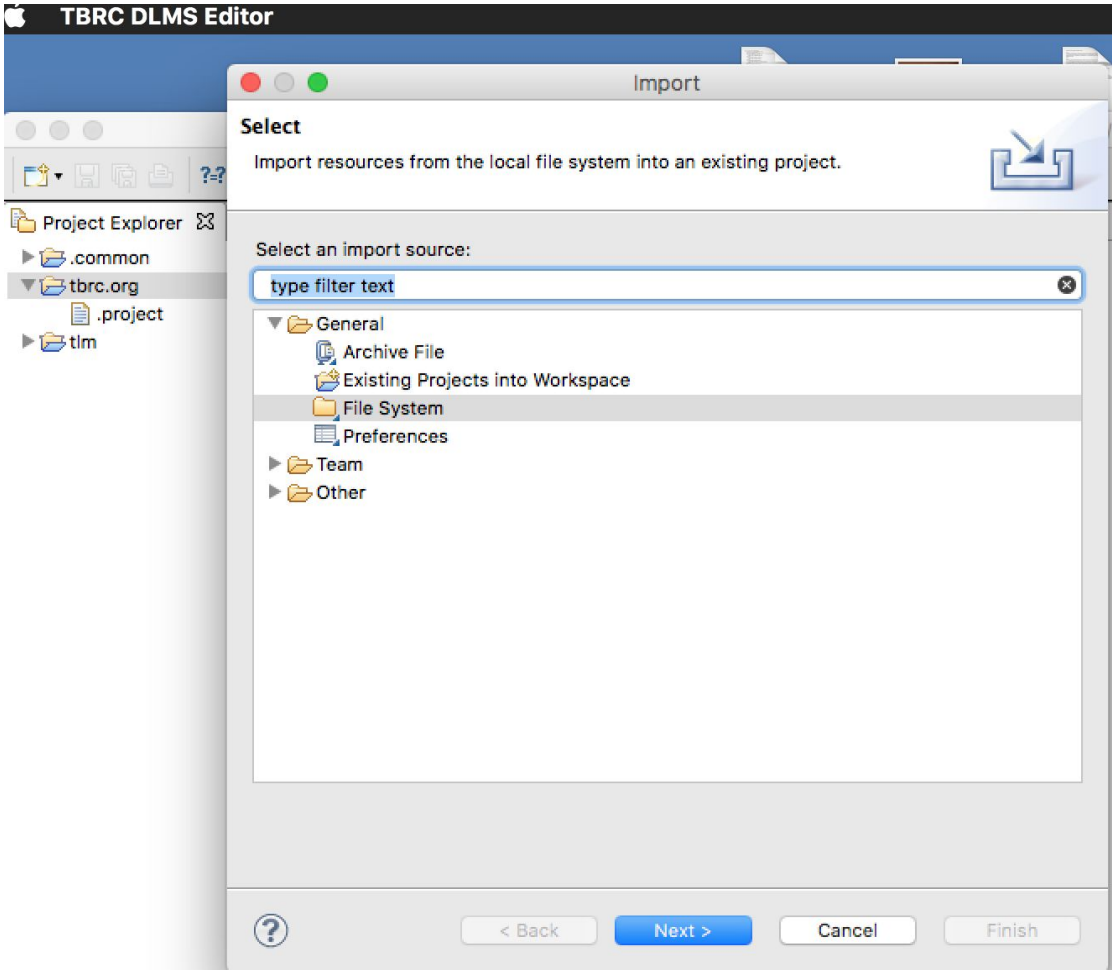
Uploading to the database:

After conversion the xml files can be uploaded to the database in two different ways. The first is the preferred way since it gives an opportunity to look at each Outline in the editor and make sure that it looks reasonable before uploading to the database:

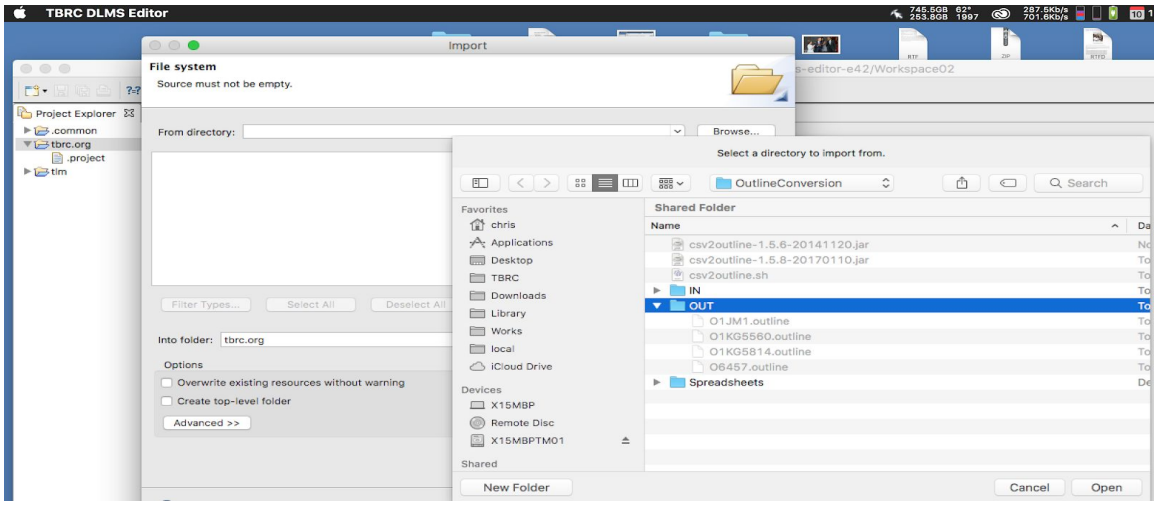
1) the DLMS Editor can be used to load each of the converted XML files and then uploading to the database. The converter names the files with `.xml` extension and in order to load into the DLMS Editor this extension must be changed to `.outline` so that the DLMS Editor recognize the document type. With the editor running navigate to **File >> Import …**:



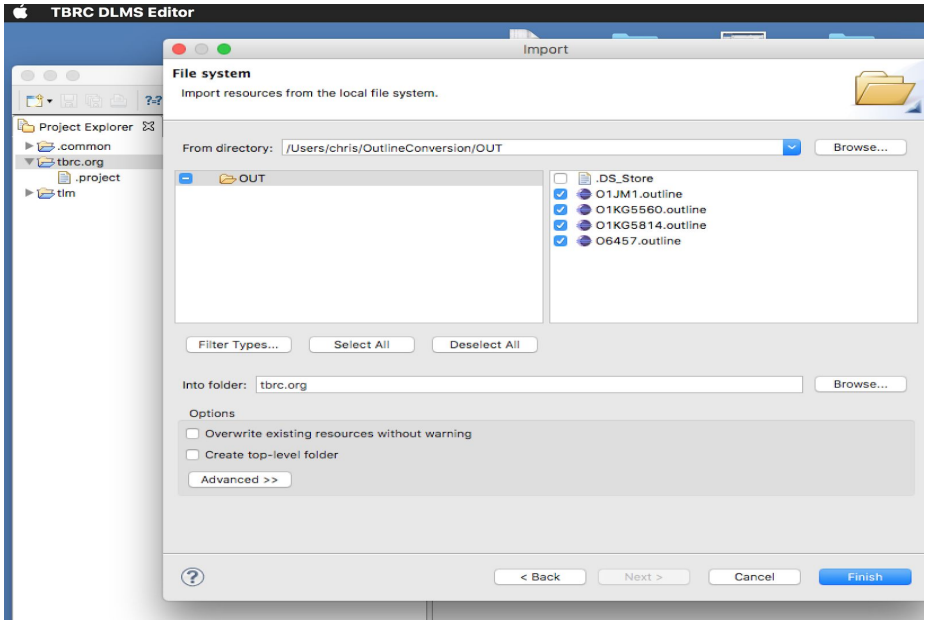
clicking on **Import…** will bring up an import dialog. Select “**File System**” and press “**Next >**”:



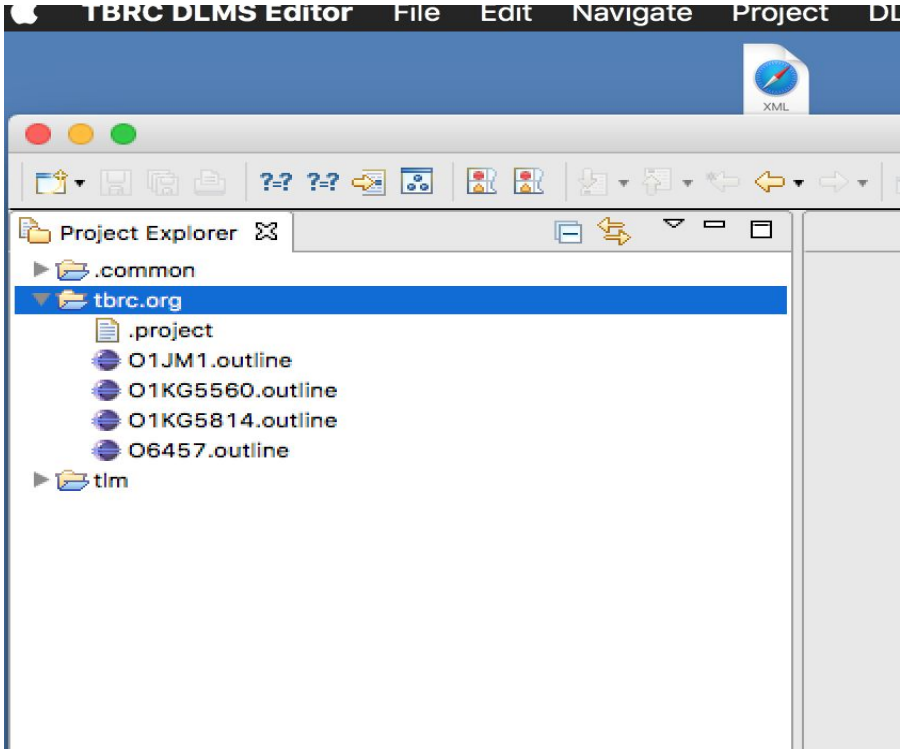
this will bring up a dialog that allows you to navigate in the file system to the folder containing the converted XML outline files. Press “Open” once you have navigated to the desired folder:



this will then open a final dialog that allows you to select the outline files that you want to import. Press “**Finish**”:



and now you should see the imported outlines in the selected project:



Now you can review each outline and when all is good then they can be uploaded via “File >>

2) by transferring the set of xml files to the server and using the eXist-db client - this method is included for completeness but requires an administrator to perform:

```
    ./bin/client.sh -u user -P pass -m /db/tbrc/tbrc-outlines -p
location-of-folder-of-outline-xml-files
```