

Creating a TEI Shell from OCR-Header from MARC or Custom Header

Note!

Files required to execute java program for TEI shell creation are maintained on this page except for project specific TEI-output stylesheets, which are maintained on the respective project wiki pages. When linking to supporting files to execute the java program, reference the files maintained on this page. Thank you!

Files Needed to Execute TEI Shell Creation

- [OCR2TEI.jar](#): Java program; execute on most DLP servers or local machine with java support
 - In most cases, java program will be executed on: //bleu.dlib.indiana.edu/digitize_data/[projectdir]/
 - You must have JRE (Java) 1.5 or later in order to run this program

For MARC-Based TEI Header

- [dataprocessing.xsl](#): Style sheet imported by generateTEIheaderFullTEI-ia.xsl; no need to update (publication date normalization)
- [MARC21slimUtils.xsl](#): Style sheet imported by generateTEIheaderFullTEI-ia.xsl; no need to update (definition of MARC tags, subtags and indicators)
- [generateTEIheaderFullTEI.xsl](#): Style sheet that generates the TEI shell with MARC metadata in the Header; expects idno parameter (e.g., VA#)
- [idno.properties](#): Properties file identifies that parameters needed by TEI Shell style sheet (attached is a sample file to be modified accordingly with corresponding VA#)

For Custom TEI Header

Documentation in progress for Custom Header output; see MARC to Header output documentation below.

This java program replaces our original Perl concatenation workflow.

- [template.xml](#) (need to be created and tested)
- not sure of the rest, need to test
- If populating from file (xml file, header derived manually), use the default.xml template that you can tailor (in the header) for specific project needs

1. Place the following files in the desired project directory on DLP server; must be level with VAA/VAB sub-directories:

- [OCR2TEI.jar](#)
- [dataprocessing.xsl](#)
- [MARC21slimUtils.xsl](#)
- [generateTEIheaderFullTEI.xsl](#) (if MARC-based Header) OR [template.xml](#) (if custom header)
- [idno.properties](#)

How the OCR to TEI Shell Program Works

- *The concatenation files need to be in the root directory that contains all the VA sub-directories for the project.*
- *It assumes there is a "OCR" (case insensitive) directory inside the VA# directory.*

```
usage: java -jar OCR2TEI.jar [-h] [-s <xslt>] -v <version> -x <xml> -n <VAA#> [-p <params>]
OCR to TEI generation tool
-h,--help                Optional. Explains arguments expected by the Java program.
-m,--metadata <arg>     If MARC-based header, the URL for the MARC metadata;
                        URL should ALWAYS be in double quotes. If custom header,
                        "template.xml" file with custom header defined.
-n,--name <arg>         VA#. This will be used as the name of the TEI file that
                        is to be generated.
-p,--params <arg>       Optional. Parameters required by the xslt that generates the
                        TEI shell. It needs to be defined in a .properties file.
                        For most projects, the VAA# is the only parameter.
-s,--xslt <arg>         Optional. Specify the XSL that will create the TEI shell. If not
                        specified, the template.xml file will be used directly.
-v,--version <arg>     Specify version of the TEI (P4|P5)
```

Executing TEI Shell for MARC-Based TEI Header

Note: This program needs to be run for each book (each VAA#). This will require updating the idno.properties file every time before running the program.

1. Make sure all the necessary files are placed in the proper directory as explained under "Setup"
2. Determine the first book that requires a TEI Shell; note the VAA#
3. Open the **idno.properties** file and update the idno value so it matches the VAA# of the book for which you are about to generate a TEI Shell XML file
 - a. Open file and edit using emacs, vi, pico or any allowable unix editor or you can open the file in notepad, update and transfer file back to directory (SFTP)
4. Open an SSH terminal
5. Run the java command
 - a. Be sure to state the proper VAA#.
 - b. Be sure you note the title control number for the book
 - i. The URL for the MARC metadata should ALWAYS be in double quotes, and is of one of two forms: Please see this page for details: [URL for MARC metadata](#)
 - ii. Be sure you updated the **idno.properties** file as specified in step 3.
 - c. All other arguments (-jar, -s, -p, -v) are the same for every book
 - d. Sample usage:
An example of using an XSLT stylesheet to transform the template during concatenation.(Please see [URL for MARC metadata](#) for an example URL)

```
java -jar OCR2TEI.jar -s generateTEIheaderFullTEI.xsl -n VAA9495 -m "http://url/to/marc" -p idno.properties -v P4
```

Another example, where only the template is used, and no XSLT.

```
java -jar OCR2TEI.jar -n VAA9495 -m p5-template.xml -v P5
```

Troubleshooting

- If you get a **Premature end of file** error, test the [MARC URL](#) with the proper title control number inserted. We have encountered problems retrieving the MARC data via SRU.

Testing the Output of TEI Shell for MARC-Based TEI Header

You should find the TEI/XML shell file level with the VAA directories.

1. After running the program once, verify that an XML file was created.
2. Check to make sure the VAAfilename.xml assigned corresponds to the intended book's VAA#.
3. Open the XML file in Oxygen
4. Make sure the file is valid
5. If file is valid, this is a good first step!
6. Spot check the bibliographic metadata
 - a. If metadata doesn't match, the title control number for the "metadata URL" is either incorrect or causing problems due to the title control number construction. Contact Jenn Riley if this is the case.
7. If file is not valid, check to make sure the proper version (P4) of the TEI was declared when running the Java program.

Testing the Output of TEI Shell for Custom TEI Header

- To be documented