**Using the ADOPT  
BBMRI-ERIC ETL Pipeline**

*Sebastian Mate, Christian Knell, Christina Schüttler, Kaisa Silander, Niina Eklund  
 and Hans-Ulrich Prokosch*

Contents

[Introduction and Overview 1](#_Toc520452654)

[Preparation & Configuration 3](#_Toc520452655)

[Getting the Latest Version 3](#_Toc520452656)

[Cleaning the Already Processed Files 3](#_Toc520452657)

[Installation of Perl and XMLStarlet 4](#_Toc520452658)

[Configuration of the Samply MDR Key-Based Authentication 4](#_Toc520452659)

[ETL Configuration 5](#_Toc520452660)

[Data Preparation (Excel Step 1) 6](#_Toc520452661)

[TablePreprocessor: Spreadsheet Table Rotator (Excel Step 2) 8](#_Toc520452662)

[MDRExtractor: MDR Metadata Extraction (Excel Step 2, Excel/VSC Step 3) 8](#_Toc520452663)

[MDRMatcher: Lexical Concept Matching (Step 4) 9](#_Toc520452664)

[MappingGUI: Editing the Mapping (Step 5) 10](#_Toc520452665)

[General Usage 10](#_Toc520452666)

[Correcting the Mapping of the Fake Data 11](#_Toc520452667)

[ETLHelper: XML Generation (Step 6) 12](#_Toc520452668)

[XML Legacy Conversion (Step 7) 14](#_Toc520452669)

## Introduction and Overview

This document describes how to use the ADOPT BBMRI-ERIC ETL tools. In order to test the tools and learn how they work, we have also provided exemplary fake biobank data which you can run through the ETL pipeline. This document can also act as a step-by-step guide for processing your own biobank data.

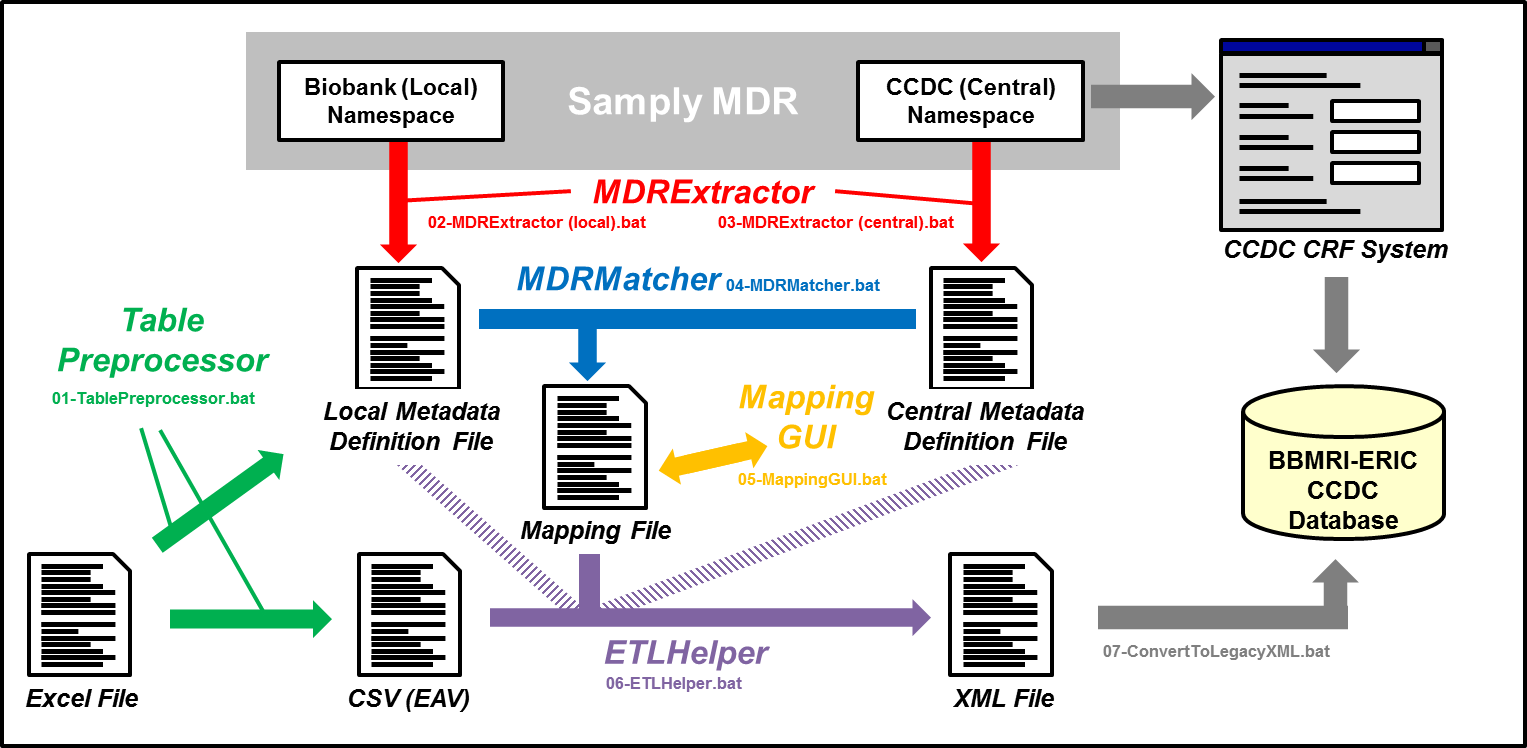
The root folder of this package contains the executable Jar files. These are small Java-based programs which are used to process the data through the ETL process.

The “Demo Biobank” directory contains the fake biobank data. Inside this directory there are various subdirectories, which are explained at later sections. The folder also contains several batch files (\*.bat), which execute various ETL-steps. These will be described later as well.

*If you want to process your own biobank data, simply duplicate the “Demo Biobank” directory and rename the copy to your preferred name.*

The following image summarizes the interaction between all tools. This manual assumes that the biobank data manager starts with an Excel file (as shown in the lower left corner) or a CSV file. This file contains all the attributes needed for the given project and which should be mapped to a target list of attributes. In addition, it contains the data type for each attribute and the actual biobank data for the given biobank donors, one row for each donor.

The ultimate goal when using the ETL pipeline is to generate the XML file in the lower right corner, which contains the processed data and can be uploaded into target research system (in the case of BBMRI-ERIC, this is the CCDC database).



Depending on whether the biobank data manager starts with an excel file, different tasks in this pipeline have to be executed, or not:

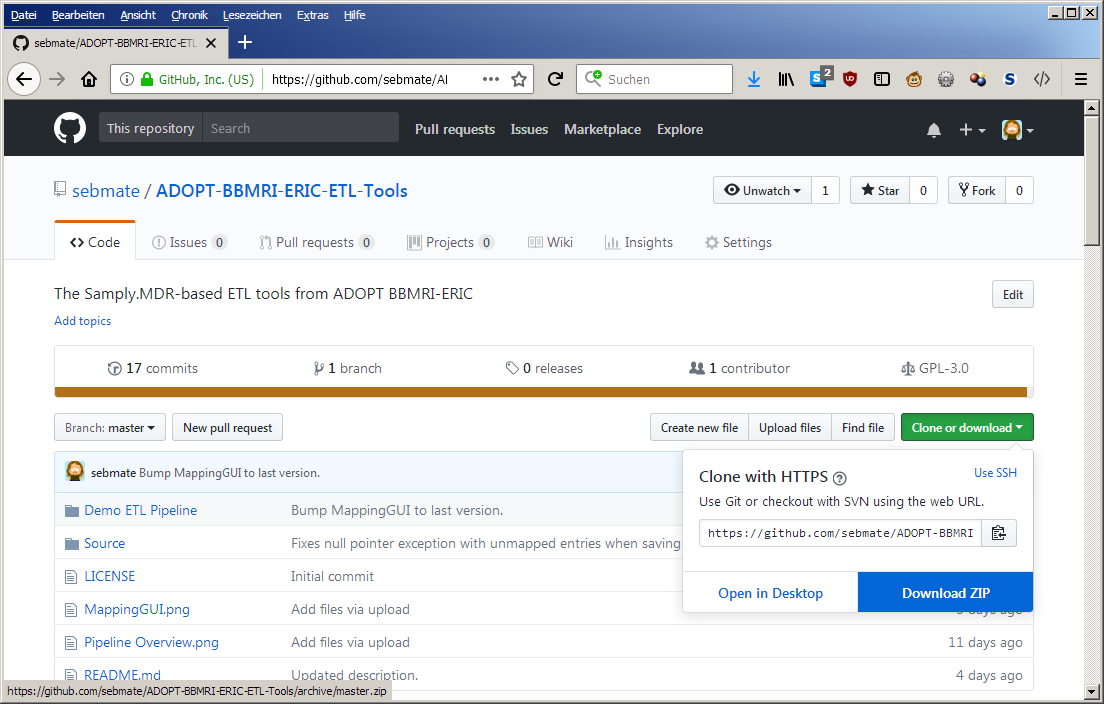
|  |  |  |
| --- | --- | --- |
| **Step** | **Excel File** | **CSV File** |
| **(Step 1)** Prepare/Define the local metadata | Annotate the Excel file so that the second row contains the data types (see “Data Preparation”) and store it as a TSV file. | Define your metadata (data elements and value sets) in the *Biobank (Local) Namespace* in the *Samply MDR*. |
| **(Step 2)** Extract the local metadata | Run **01-TablePreprocessor.bat**. This also converts the Excel data into the CSV file format (therefore the following steps are the same on both sides). | Run **02-MDRExtractor (local).bat** |
| **(Step 3)** Extract the central metadata (\*) | Run **03-MDRExtractor (central).bat** | Run **03-MDRExtractor (central).bat** |
| **(Step 4)** Perform the lexical matching with MDRMatcher | Run **04-MDRMatcher.bat** | Run **04-MDRMatcher.bat** |
| **(Step 5)** Edit the mappings with the MappingGUI | Run **05-MappingGUI.bat** and manually curate the mappings | Run **05-MappingGUI.bat** and manually curate the mappings |
| **(Step 6)** Transform the data into the XML format with ETLHelper | Run **06-ETLHelper.bat** | Run **06-ETLHelper.bat** |
| **(Step 7)** Convert the XML file into the old XML format | Run **07-ConvertToLegacyXML.bat** | Run **07-ConvertToLegacyXML.bat** |

(\*) Note that this step has to be executed only once. The central metadata is also already included in the downloaded package, so this step can be skipped. It is also necessary to have properly configured the MDR authentication for this to work.

## Preparation & Configuration

### Getting the Latest Version

To get the latest version of the ETL package, please go to <https://github.com/sebmate/ADOPT-BBMRI-ERIC-ETL-Tools>. Please note that an updated version of this manual can be found at the repository.



To download the package, please click on the green “Clone or download” button, and then select “Download ZIP”. Your browser will download the ZIP file. After uncompressing this file, you will find the demo pipeline in the “Demo ETL Pipeline”.

### Cleaning the Already Processed Files

This package already contains the supposed results from the various steps. If you want to start with a “clean” installation, please execute **00-Clean.bat**. The ultimate goal when following this guide is to create the file **xmlFiles\Legacy.xml**.

### Installation of Perl and XMLStarlet

*This step is optional and not necessarily required if you want to process the supplied fake data.*

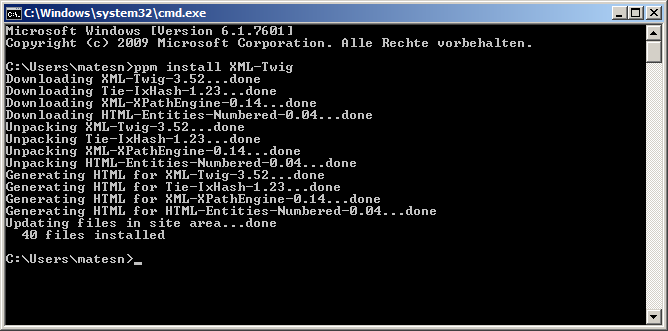
The last step in the ETL pipeline is the conversion of the generated XML file into a legacy XML schema and its validation against the schema (as triggered in **07-ConvertToLegacyXML.bat**). This “legacy” schema is an older XML file format that is required for the CCDC import. This step takes care of the conversion into this older format.

For the conversion, a Perl script is used (**convert-to-legacy-XML.pl**). To be able to run this step, you need to install *Perl* (see <https://www.perl.org/>).

It is recommended to install *Strawberry Perl* distribution, already as it contains the required additional modules.

If you install the *ActivePerl* distribution, you need to install XML-Twig. You can do this by opening a command line window and entering:

ppm install XML-Twig



To enable the validation against the XML schema, put **xml.exe** from the XMLStarlet software (see: <http://xmlstar.sourceforge.net/>) into the root directory of this package.

### Configuration of the Samply MDR Key-Based Authentication

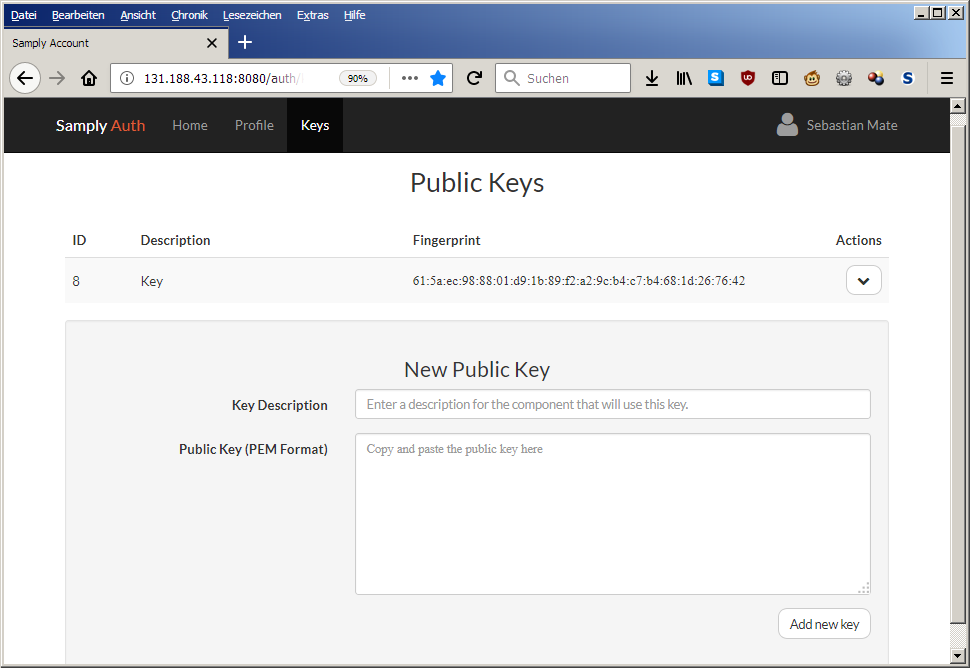
*This step is not necessary if you want to process Excel data or the supplied fake data. It is only required if you have described your metadata in the Samply MDR.*

To enable the MDRExtractor tool to access the Samply MDR, an authentication procedure needs to be configured. First, create a new pair of keys (**private.key** and **public.key**) with the following commands on a Linux machine:

openssl genrsa -out /tmp/bbmri 4096 -days 3650  
openssl rsa -in /tmp/ bbmri -pubout > public.key  
openssl pkcs8 -topk8 -in /tmp/ bbmri -outform DER -nocrypt | base64 > private.key

Second, open the Samply Auth instance running in Erlangen (and which is used for the Samply MDR). You can access this instance at <http://131.188.43.118:8080/auth/>.

Log in with your user and in the upper menu, and then select “Keys”. Now copy the contents of the public key (located in the **public.key** file, which you had generated above) into the form and provide a description (e.g. “Key”). Clicking on the “Add new key” should result in a new entry in the table, as shown in the screenshot below:



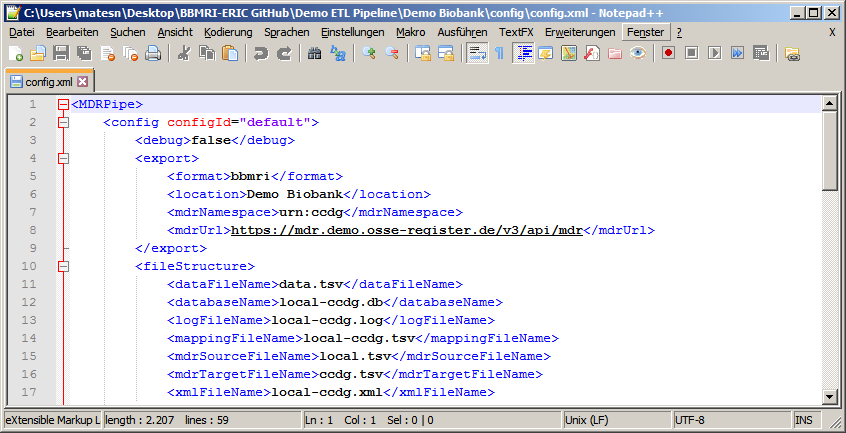
### ETL Configuration

*This step is required only if you want to process real biobank data.*

To define various parameters related to the ETL process, open the file **config\config.xml** with a text editor. These parameters include e.g. the name of your biobank or where all files are located.

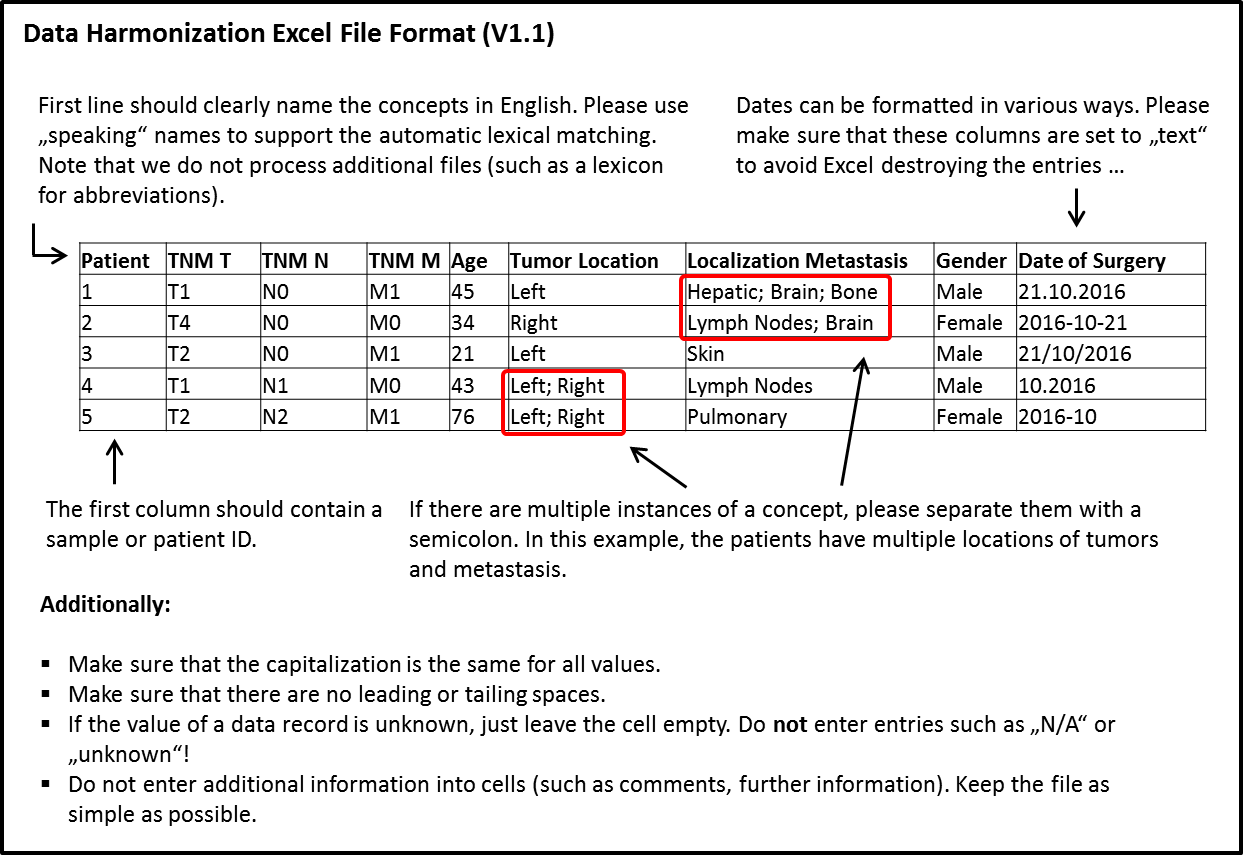
Replace the entry “Demo Biobank” in the tag <location> with your biobank’s name.

If necessary, enter your MDR keys into the tags <publicKey> and <privateKey>. This will later enable the MDRExtractor to access the Samply MDR.

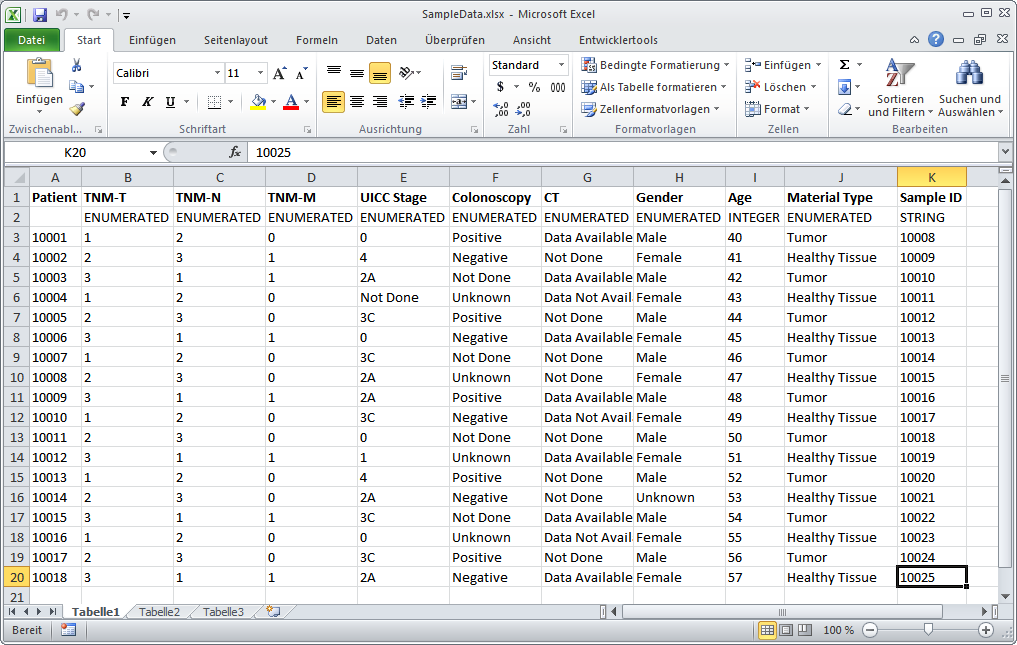


## Data Preparation (Step 1 Excel)

We assume that your source data (to be integrated with this pipeline) has been formatted according to the description provided by BBMRI-ERIC, as shown in the picture below:

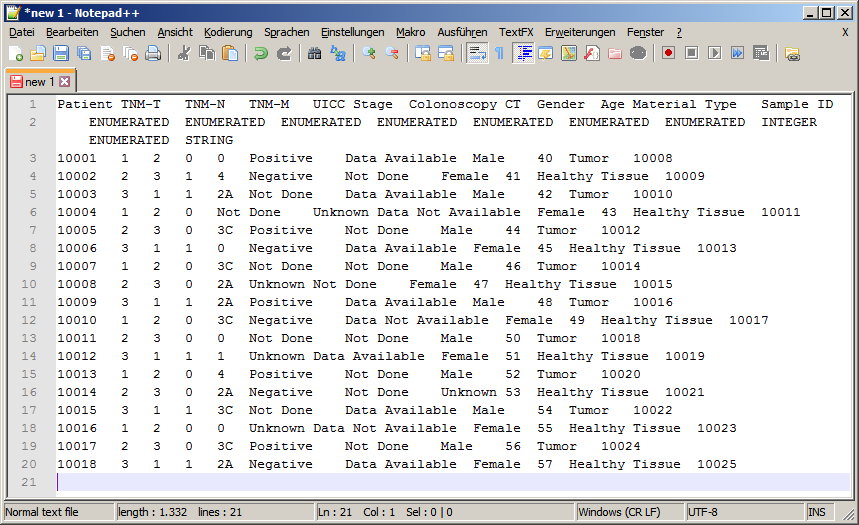


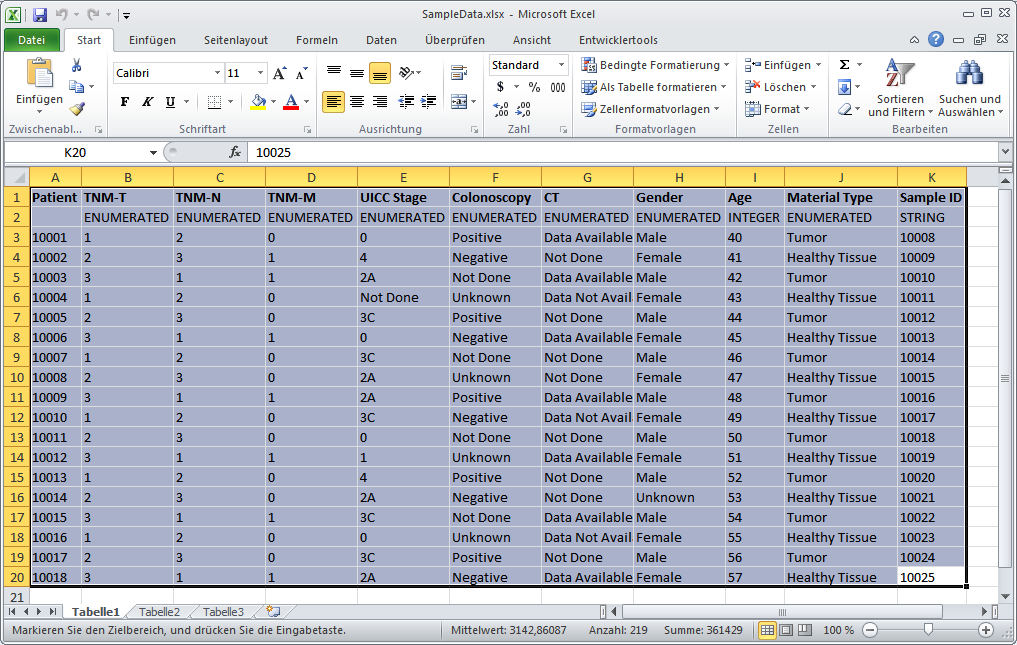
A sample Excel file is provided in this package under **dataExcel\SampleData.xlsx**:

  
As you may have observed, the second row contains data types for the columns (“ENUMERATED”, “INTEGER”, “STRING”).

If you are about to process your own data, you have to add such data types for each column (in the second row). The allowed data types (according to the Samply MDR) are:

* ENUMERATED
* INTEGER
* FLOAT
* BOOLEAN
* STRING
* DATE
* DATETIME

Next, copy the contents of the Excel spreadsheet into a text file. This automatically creates a TSV file, where each column is separated with a tabulator character.



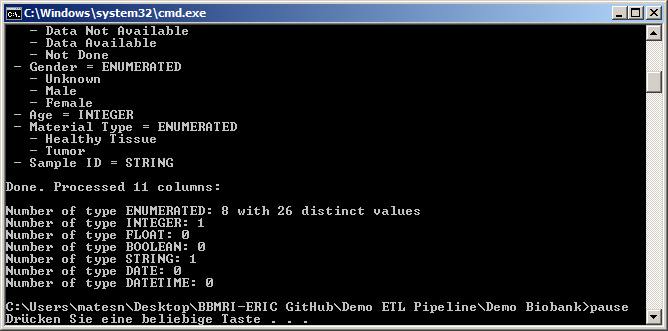
Save this file under **dataExcel\excel.tsv**.

## TablePreprocessor: Spreadsheet Table Rotator (Step 2 Excel)

*This step is only required if you want to process the supplied fake data or any other Excel-based source data. It is not required if you are using the approach based on the EAV format and the Samply MDR (not described here in detail).*

The next step is to run the TablePreprocessor tool, which rotates the Excel data from the column-based format into an EAV-like format and also generates the local metadata definition file (see first image of this document).

To run the tool, please execute **01-TablePreprocessor.bat**.



The tool displays information during execution. If you observe that the value sets are wrong, then you may need to change the data types in the Excel file.

After execution, the EAV formatted data is located in **dataEAV\data.tsv**. Similarly, the local metadata definition file is stored in **metadata\local.tsv**.

## MDRExtractor: MDR Metadata Extraction (Step 2 Excel, Step 3 Excel&CSV)

*This step is not required if you want to process the supplied fake data.*

The MDRExtractor tool is used for extracting the metadata from the Samply MDR.

To extract the local (source) metadata, please execute **02-MDRExtractor (local).bat**. The local metadata will be stored in **metadata\local.tsv**. This step will contact the MDR that has been configured as the *source MDR* in the **config\config.xml** file (see respective section above).

***Important:*** This step is only necessary if you are using the approach based on the CSV format and the Samply MDR. It also requires a proper configuration of the Samply MDR key authentication (see above) and requires you to have entered your metadata into the Samply MDR. Do not run this step if you’re using Excel files as input, as this will overwrite the metadata definition file that was derived with the TablePreprocessor tool.

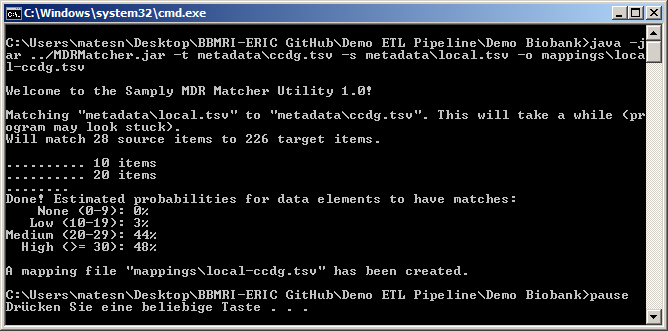
To extract the central (target) metadata, please execute **03-MDRExtractor (central).bat**. The central metadata will be stored in **metadata\ccdg.tsv** (as configured in the configuration file).This step will contact the MDR that has been configured as the *target MDR* in the **config\config.xml** file (see respective section above).

***Important:*** This step will extract the very last versions of the metadata from the Samply MDR, something that is not necessarily desired! For example the ADOPT BBMRI-ERIC CCDC actually uses an older version of the metadata. Ask the BBMRI-ERIC support what to use if you’re unsure.

## MDRMatcher: Lexical Concept Matching (Step 4)

The next step is to execute the MDRMatcher tool, which compares the source metadata with the target metadata and tries to create mappings.

To run the tool, please execute **04-MDRMatcher.bat**.



After execution, the mapping information is stored in **mappings\local-ccdg.tsv**. A logging file is stored in **mappings\local-ccdg.tsv.log**. The logging file allows one to understand how the MDRMatcher came up with its mapping suggestions.

**Translating the Metadata into English**

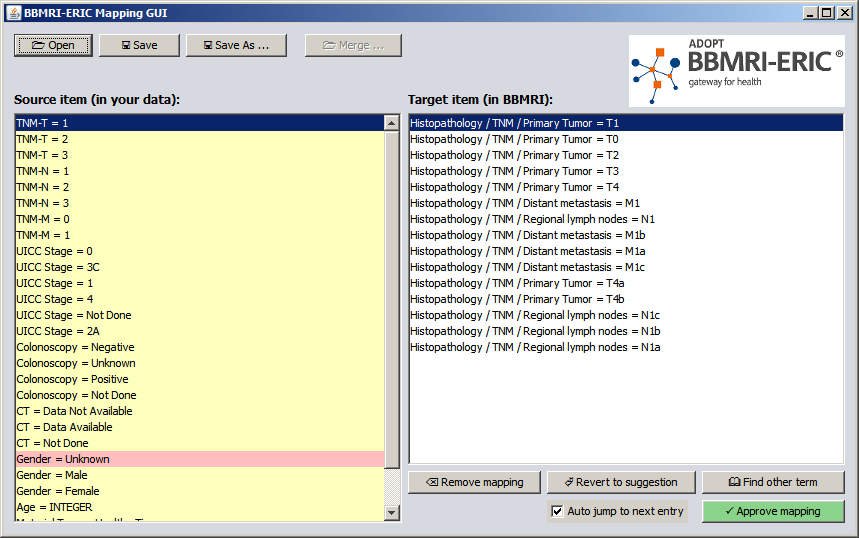
*This step is not required if you want to process the supplied fake data.*

If your source data is in a language that is different to the language in the target terminology (which is English), you can translate the right-most column in your source metadata file (**metadata\local.tsv**). This will significantly increase the quality of the matching result.

## MappingGUI: Editing the Mapping (Step 5)

The next step is to verify (or, if necessary, to correct) the mappings suggested by MDRMatcher. This is the only step that requires work by the user (i.e., you). We have tried to make this process as smooth and simple as possible by providing a graphical user interface for this task with the MappingGUI program.

To run the tool, please execute **05-MappingGUI.bat**.



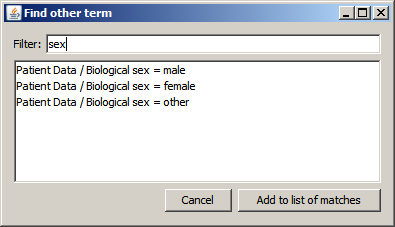
### General Usage

The tool displays the data elements and values from your source data (aka Excel file) on the left, and the proposed mappings on the right. Note that the mapping is done on the “data value” level, not on the data item level.

Entries on the left are shown in red when no mapping has been proposed by MDRMatcher. They are displayed in yellow, if a mapping has been proposed but was not approved by the user. The mapping will turn green after it has been approved by the user.

Your task is to go through the left list of items and verify that the proposed mapping is the correct one. The proposed mapping is the selected entry in the list. If nothing is yet selected or if the proposal is wrong, you simply have to select another one by clicking on it.

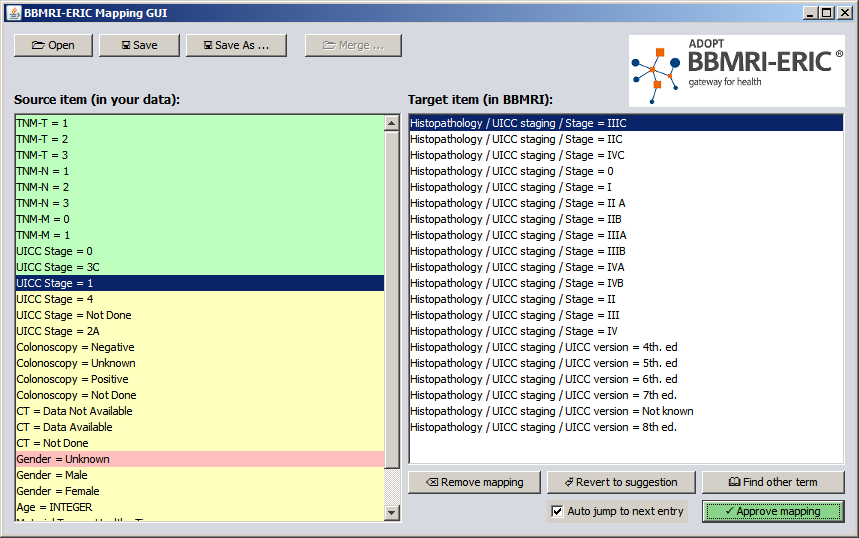
If the proposed mappings on the right side do not contain the proper entry, it can be searched manually by clicking on the button “Find other term”. A search window will open where you can search the target terminology. To add an entry to the proposed mappings in the main window, simply double-click on it or use the button “Add to list of matches”.



### Correcting the Mapping of the Fake Data

This section will demonstrate using the MappingGUI program by continuing with the supplied fake data of the “Demo Biobank”.

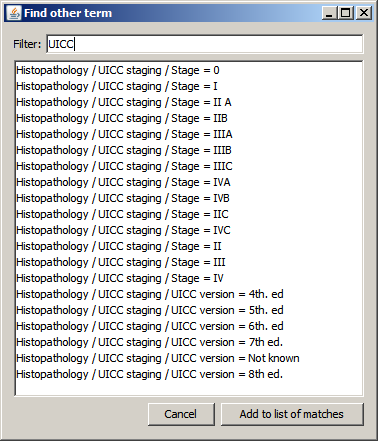
As shown in the screenshot above, the first proposal is a correct one, as it describes a mapping between “TNM-T = 1” and “Histopathology / TNM / Primary Tumor = T1”. To approve this mapping, click on “Approve mapping”. The entry in the left column will turn green (approved). The next proposed mappings are also correct. Continue with the next entries until you reach “UICC Stage = 1”:



As it can be seen, the proposed mapping between “UICC Stage = 1” and “Histopathology / UICC staging / Stage = IIIC” is wrong. To correct it, select the correct entry and press “Approve mapping”.

Also do this with the entry “UICC Stage = 4”.

For the entry “UICC Stage = Not Done”, there is only a proposal of “Not known”, which is something different (“Unknown” is not the same as “Not done”!). To search for other UICC values, click on “Find other term” and enter “UICC” into the filter field:



As it can be seen, there is no entry with the value “Not done”. So the proper way of handling this entry would be to remove the current mapping. Click on “Cancel” to close the search window. In the main window, click on “Remove mapping” and then “Approve mapping”.

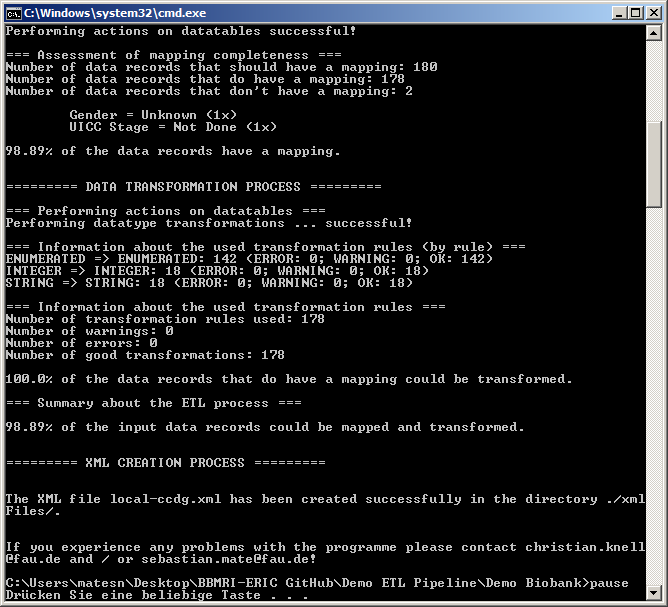
A similar mapping is the one for “Gender = Unknown”. There is no corresponding entry in the target terminology (you can verify this by using the “Find other term” button), so remove the mapping and then approve it. Note that this does not totally remove the “Unknown” entry from the mapping file (it therefore stays in the left window), but pressing the Approve button stores the information that there is no mapping for this value (“it is approved that there is no mapping”). That’s why the entry turns green, even if no mapping has been selected on the right side.

Continue with the remaining mapping suggestions, then click on “Save” and close the program. Note that you can close the program after saving everything and resume your work later.

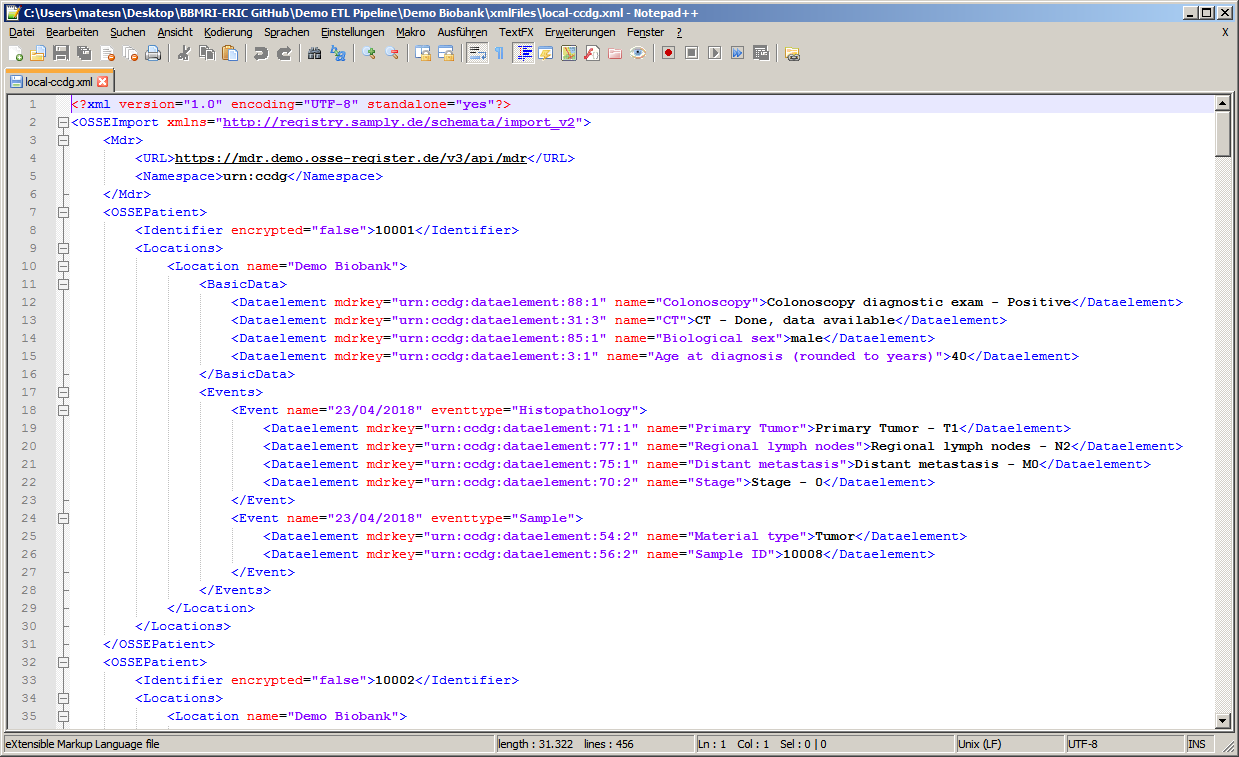
## ETLHelper: XML Generation (Step 6)

The data are now ready to be transformed into the XML file, which can be uploaded into the BBMRI-ERIC target system (e.g. the CCDC system).

To run the tool, please execute **06-ETLHelper.bat**.



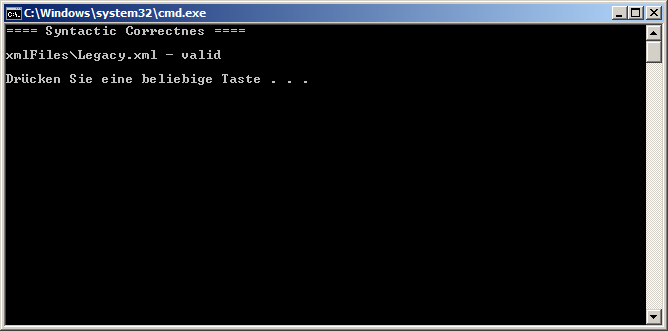
After execution, the generated XML file can be found in **xmlFiles\local-ccdg.xml**. Note that this XML file complies with the “new” XML format which has been developed in December 2017. In order to transform it into the “old” (legacy) format, please continue with the next step.



## XML Legacy Conversion (Step 7)

To convert the generated XML file into the legacy format, please make sure you put **xml.exe** from the XMLStarlet software (see: http://xmlstar.sourceforge.net/) into the root directory of this package.

To start the conversion, please execute **07-ConvertToLegacyXML.bat**.



The script will also validate the XML file. If this succeeds, it will display “valid”, as in the screenshot above. If not, it will show “invalid”.

The generated XML file can be found in **xmlFiles\Legacy.xml**.

