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

*Sebastian Mate, Christian Knell, Christina Schüttler and Hans-Ulrich Prokosch*

Contents

[Introduction and Overview 1](#_Toc512252747)

[Preparation & Configuration 2](#_Toc512252748)

[Getting the Latest Version 2](#_Toc512252749)

[Installation of Perl and XMLStarlet 3](#_Toc512252750)

[Configuration of the Samply MDR Key-Based Authentication 3](#_Toc512252751)

[ETL Configuration 4](#_Toc512252752)

[Data Preparation 5](#_Toc512252753)

[TablePreprocessor: Spreadsheet Table Rotator 7](#_Toc512252754)

[MDRExtractor: Getting the Data out of the MDR 7](#_Toc512252755)

[MDRMatcher: Lexical Concept Matching 8](#_Toc512252756)

[MappingGUI: Editing the Mapping 8](#_Toc512252757)

[General Usage 9](#_Toc512252758)

[Correcting the Mapping of the Fake Data 10](#_Toc512252759)

[ETLHelper: XML Generation 11](#_Toc512252760)

[XML Legacy Conversion 13](#_Toc512252761)

## Introduction and Overview

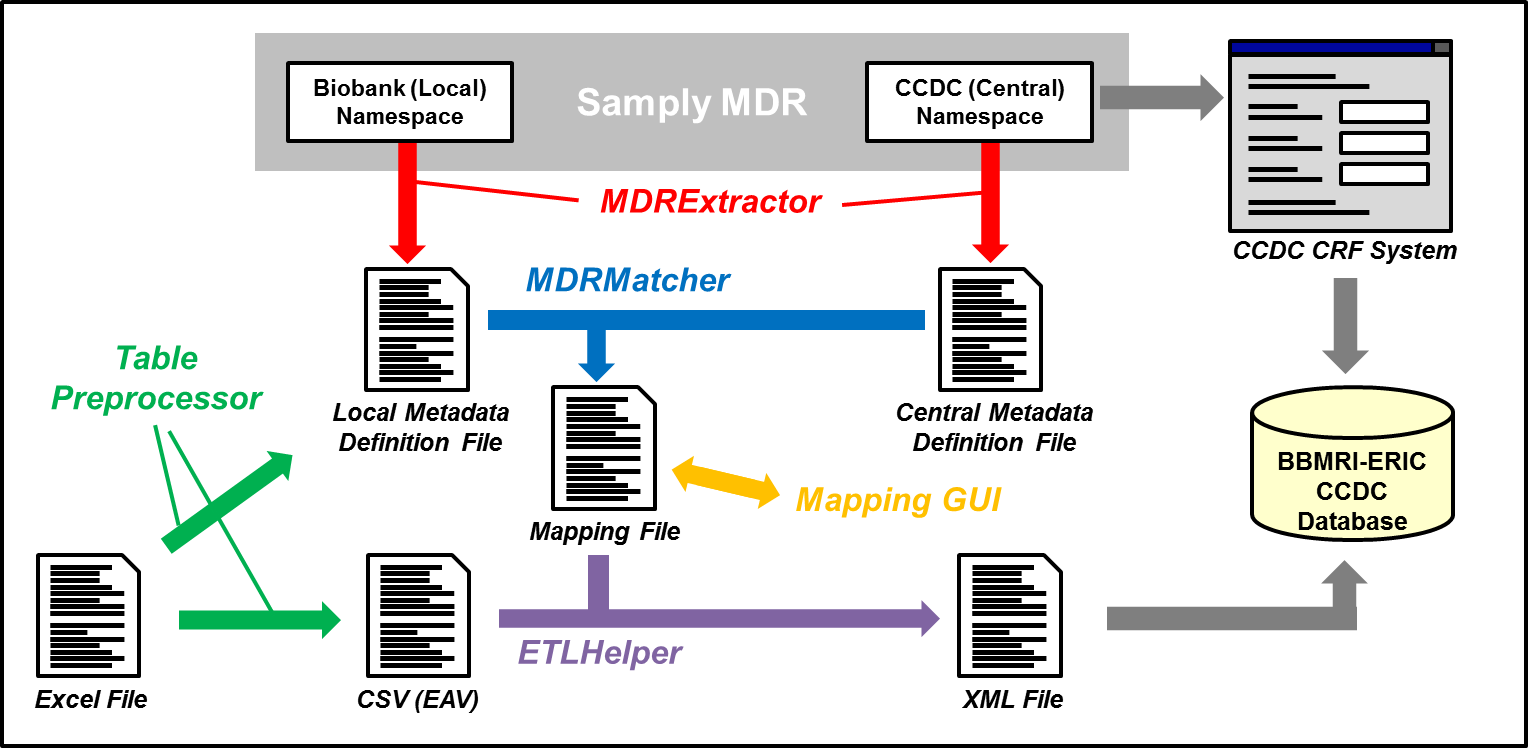
This document describes how to use the ADOPT BBMRI-ERIC ETL tools. It does this by running exemplary fake biobank data, which are also included in this package, through the pipeline. This document can also act as a step-by-step guide for processing your own patient data.

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

The “Demo Biobank” directory contains the fake biobank data. Inside this directory there are various subdirectories, which we’ll cover later. 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 data, simply duplicate the “Demo Biobank” directory and rename the copy to your preferred name.*

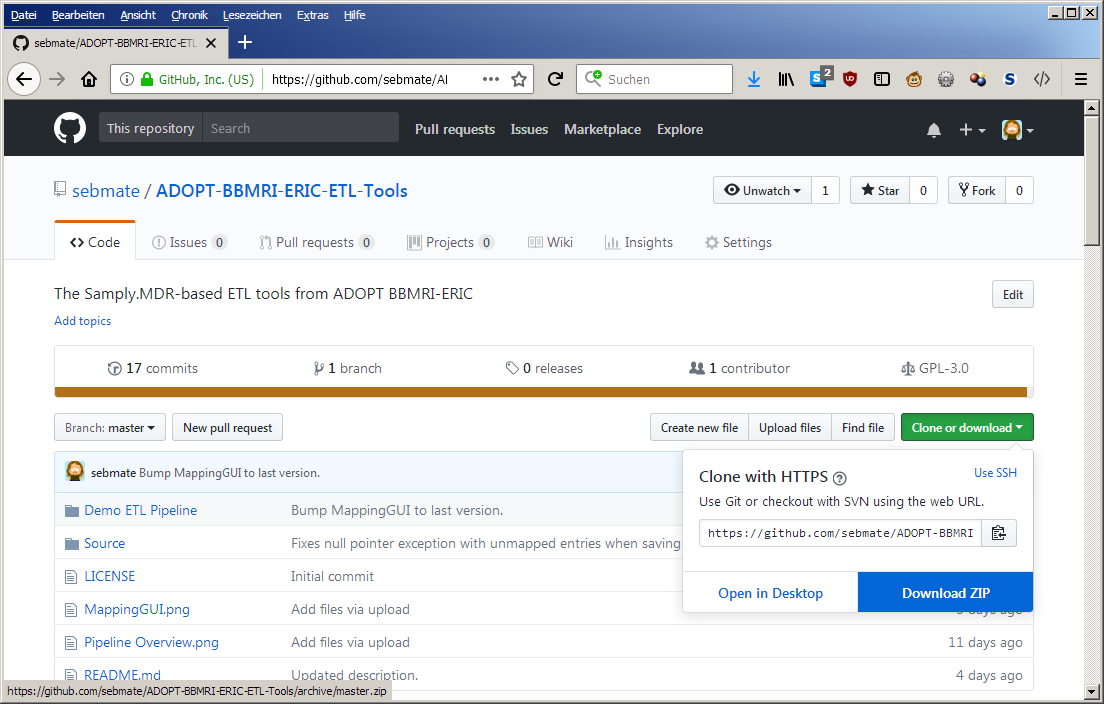
The following image summarizes the interaction between all tools. Document assumes that the reader starts with an Excel file (as shown in the lower left corner). The goal of the ETL pipeline is to generate the XML file in the lower right corner.



## Preparation & Configuration

### Getting the Latest Version

To get the last version of this package, please go to <https://github.com/sebmate/ADOPT-BBMRI-ERIC-ETL-Tools>. Please note that this may also include an updated version of this document!



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”.

### 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**).

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. Please follow the documentation on <https://www.perl.org/> or seek advice from your IT department.

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

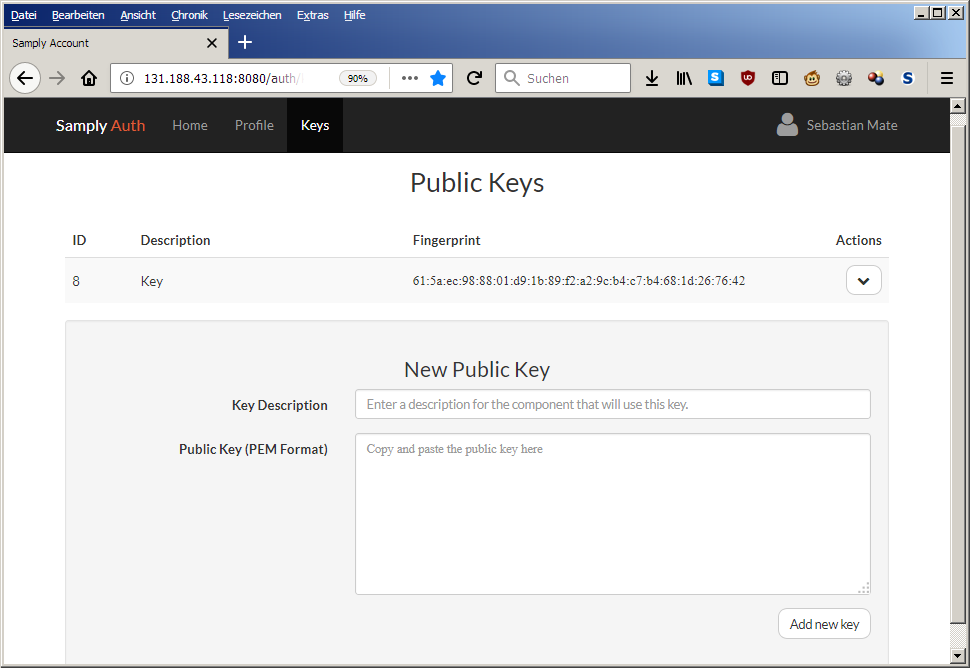
*This step is not necessary if you want to process Excel data or the supplied fake data. It is only required if you have described your metadata in the Samply MDR.*

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:



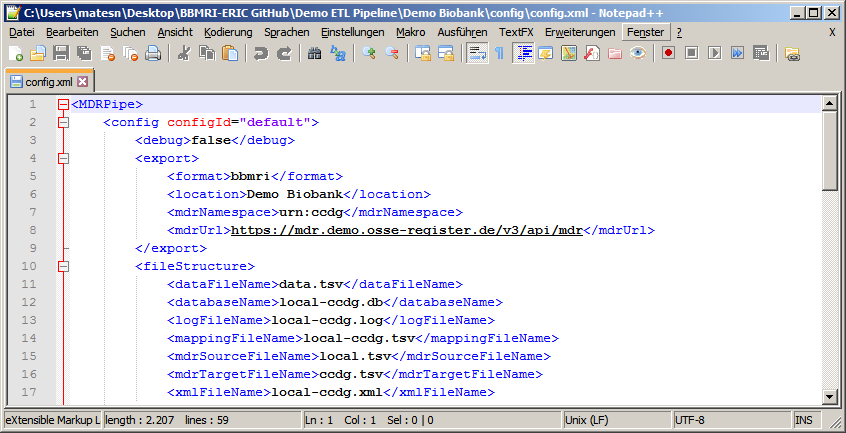
### ETL Configuration

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

To define various parameters related to the ETL process, open the file **config\config.xml** with a text editor.

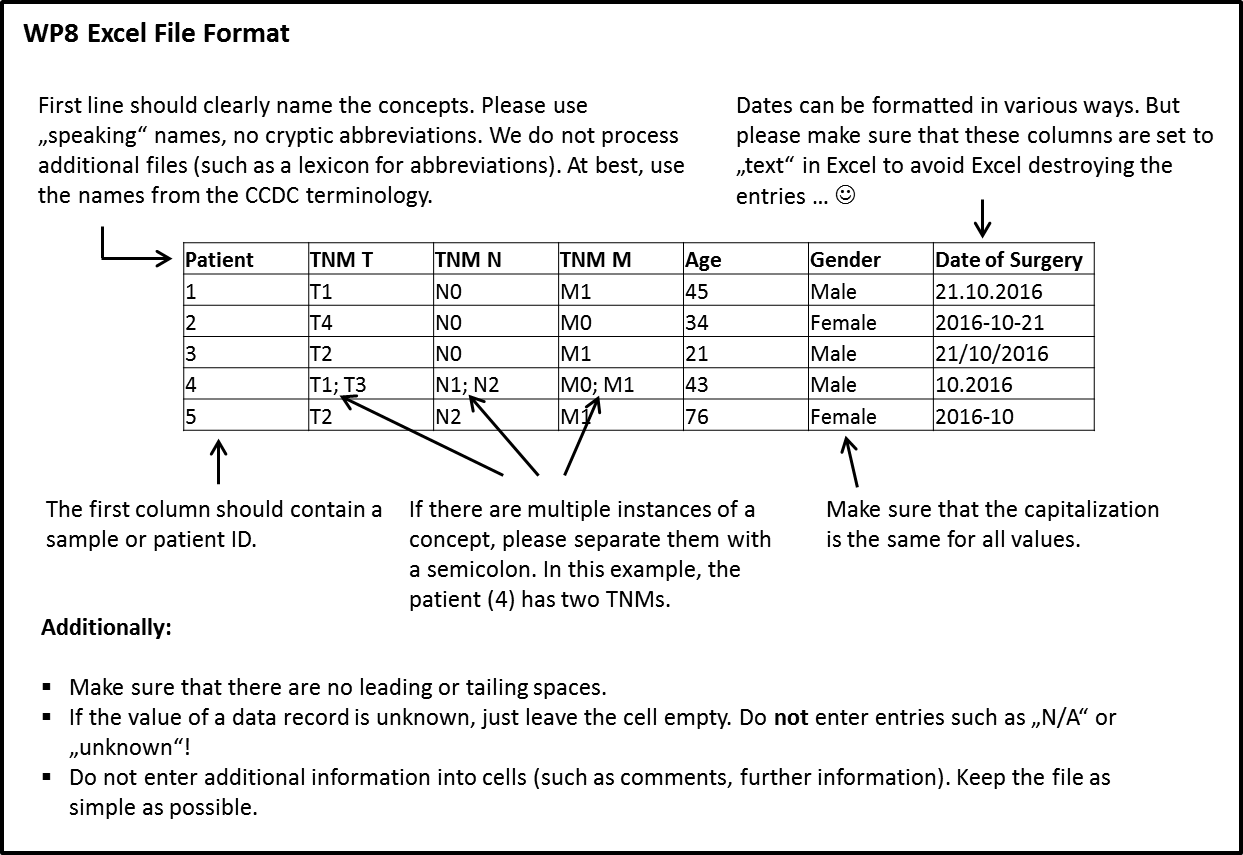
Replace the entry “Demo Biobank” in the tag <location> with your biobank’s name.

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

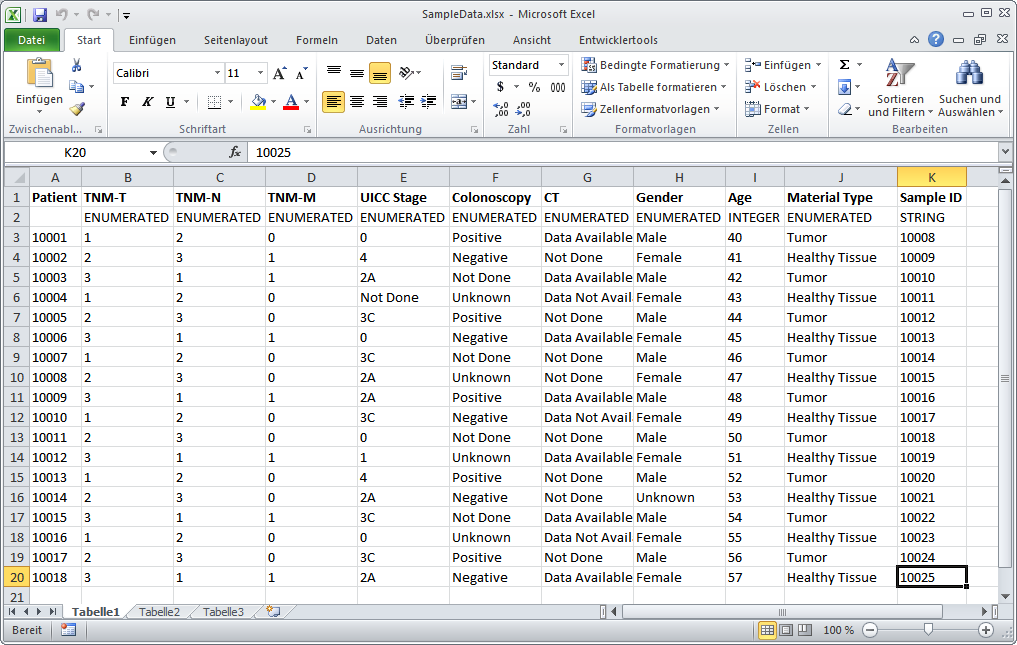


## Data Preparation

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:

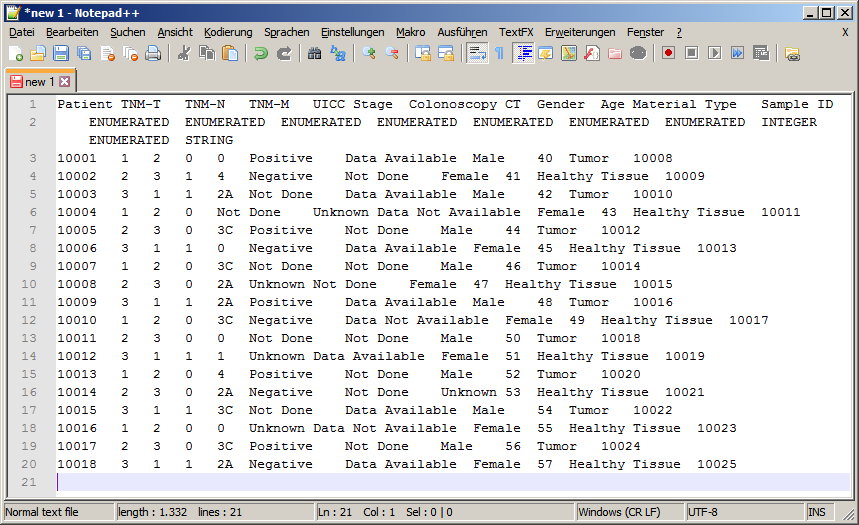


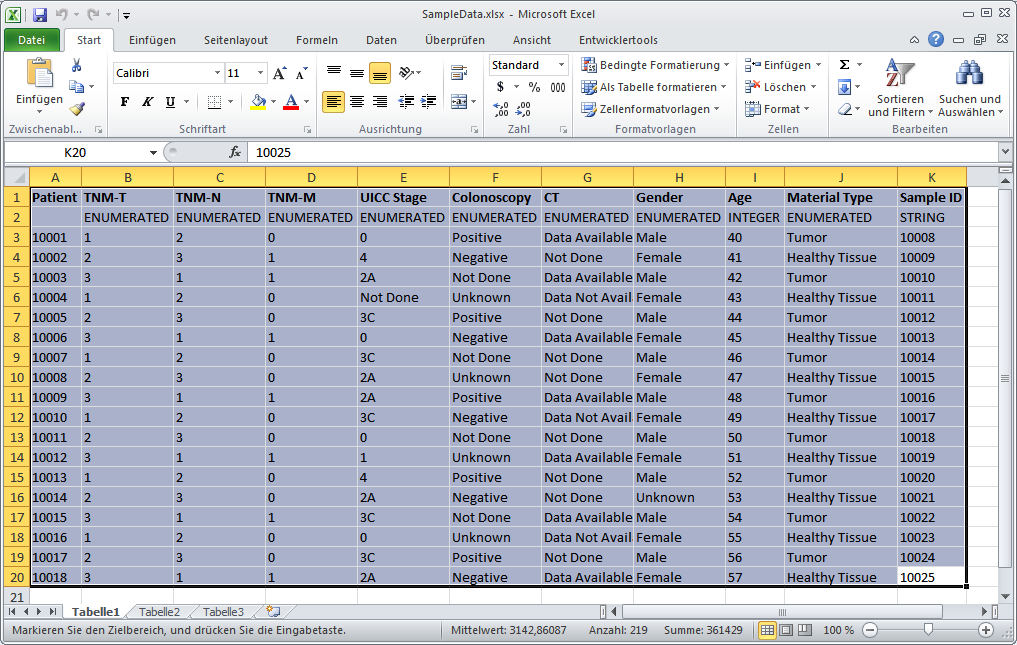
A sample Excel file is provided in this package under **dataExcel\SampleData.xlsx**:

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

If you’re 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 (tabulator separated value) file.



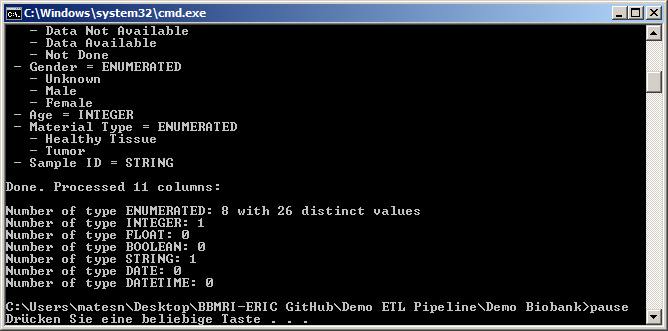
Save this file under **dataExcel\excel.tsv**.

## TablePreprocessor: Spreadsheet Table Rotator

*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: Getting the Data out of the MDR

*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**.

***Important:*** This step is only necessary if you are using the approach based on the EAV 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.

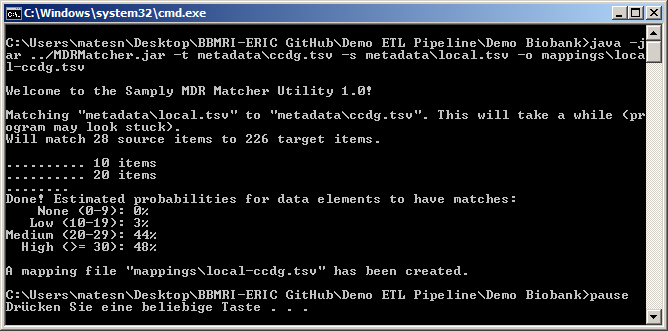
To extract the central (target) metadata, please execute **03-MDRExtractor (central).bat**. The central metadata will be stored in **metadata\ccdg.tsv** (as configured in the configuration file).

***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

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**. This allows one to understand how the MDRMatcher came up with its mapping suggestions.

**Translating the Metadata into English**

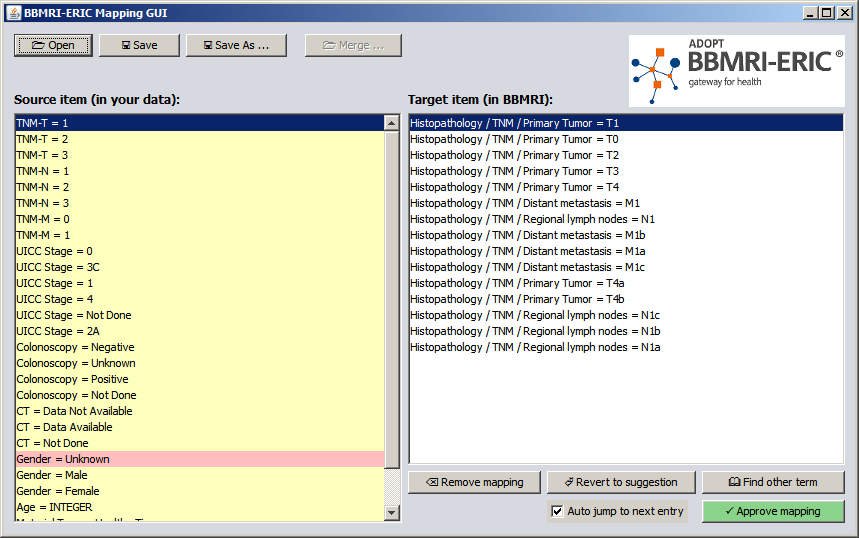
*This step is not required if you want to process the supplied fake data.*

If your source data is in a language that is different to the language in the target terminology (which is English), you can to 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

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’ve 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**.



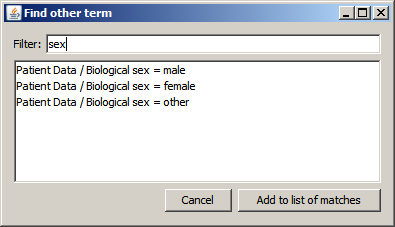
### General Usage

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.

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.

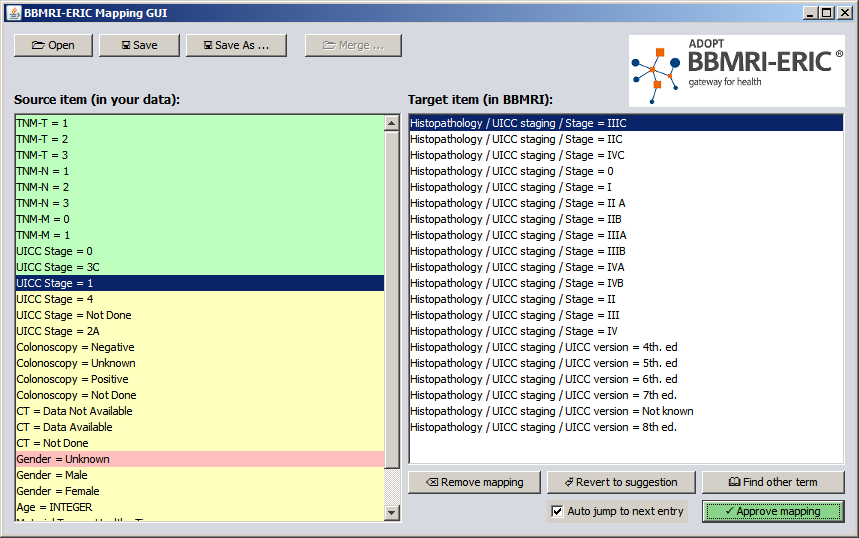
If the proposed mappings on the right side do not contain the proper entry, it can be searched manually by clicking on the button “Find other term”. A search window will open where you can search the target terminology. To add an entry to the proposed mappings in the main window, simply double-click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

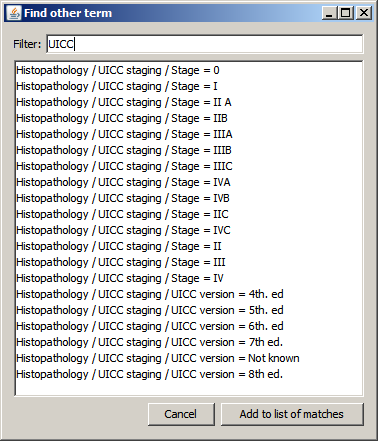
As shown in the screenshot above, the first proposal is a correct one, as it describes a mapping between “TNM-T = 1” and “Histopathology / TNM / Primary Tumor = T1”. To approve this mapping, click on “Approve mapping”. The entry in the left column will turn green (approved). The next proposed mappings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

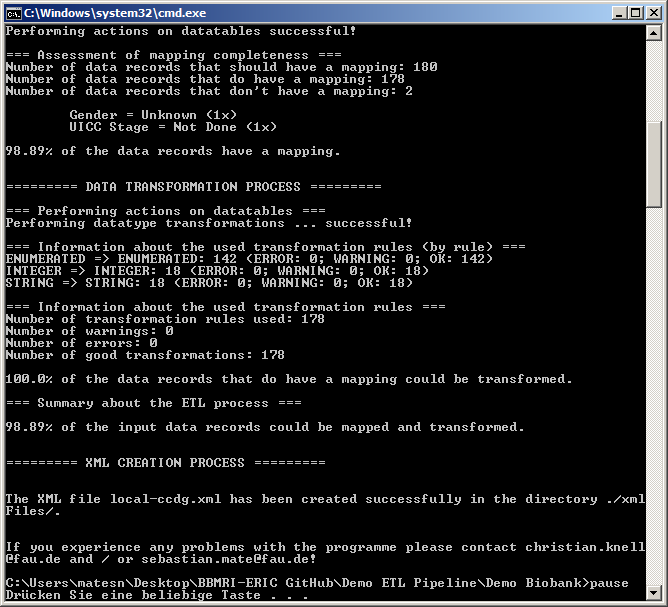
A similar mapping is the one for “Gender = Unknown”. There is no corresponding entry in the target terminology (you can verify this by using the “Find other term” button), so remove the mapping and then approve it.

Continue with the remaining mapping suggestions, then click on “Save” and close the program.

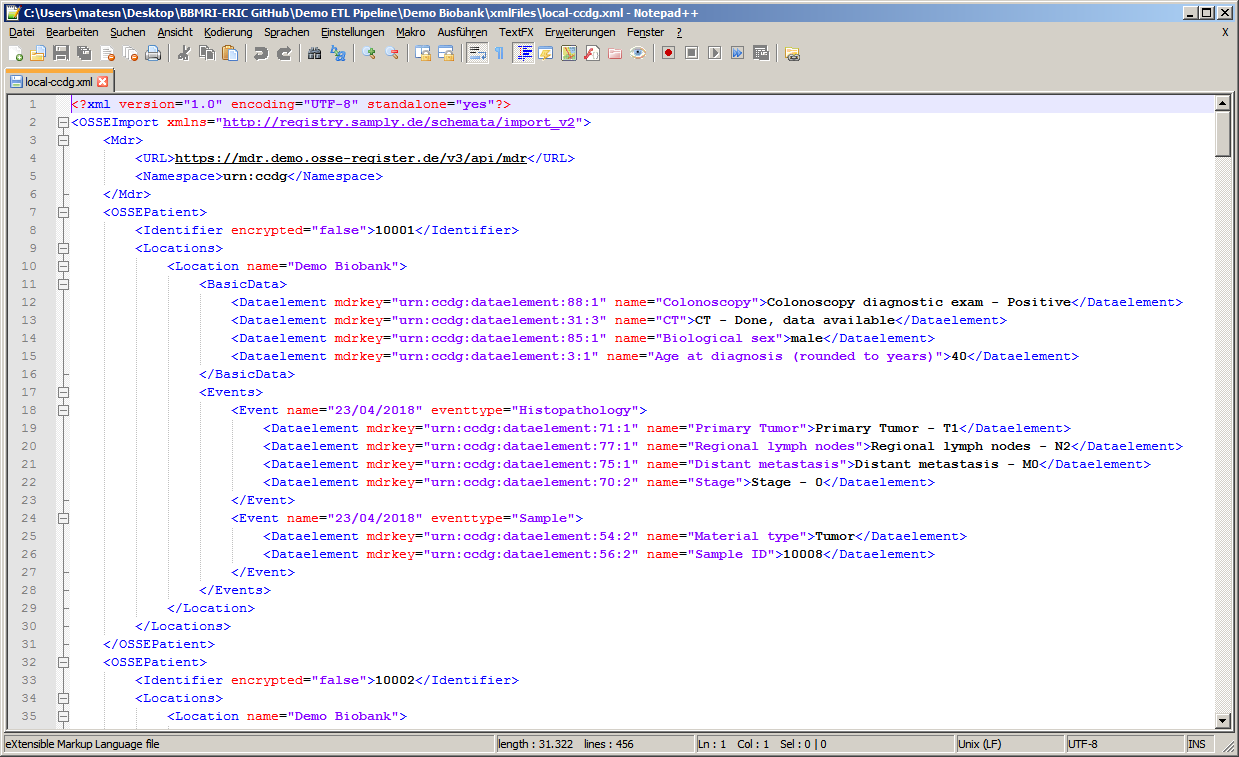
## ETLHelper: XML Generation

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**.



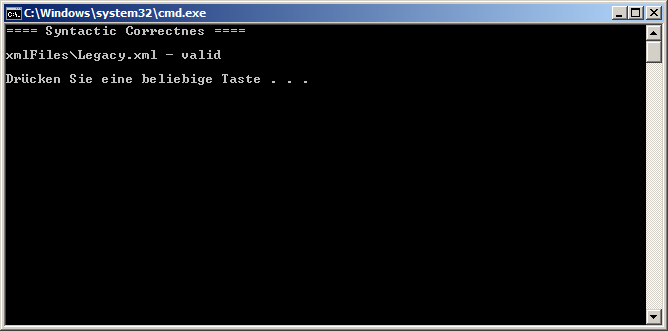
After execution, the generated XML file can be found in **xmlFiles\local-ccdg.xml**. Note that this XML file complies with the “new” XML format was has been developed in December 2017. In order to transform it into the “old” (legacy) format, please continue with the next step.



## XML Legacy Conversion

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

