Instructions for MARCXML derived files (Kadoc and Ancient books)

The files are a csv version of the original MARCXML files, encoding information in the fields described by the MARC21 model. Make sure you have a look at the model before reading the rest of the document.

When you upload the document, make sure that you configure the right parsing options:

1. commas (CSV)
2. Trim leading & trailing whitespace from strings. Escape special characters with \
3. Parse next 1 line as column headers
4. Use character " to enclose cells containing column separators
5. Store blank rows, store blank cells as nulls

In the passage from XML to CSV, the following conventions apply:

1. The field number is used as header of the column
2. the first indicator, second indicator and subfield are stored in the Cell, separated by a $c
3. To mark the beginning of a new field, I have used the ^symbol, preceding the first $

Examples:

- The XML structure:

```
<datafield tag="035" ind1=" " ind2=" ">
        <subfield code="a">(BeLVLBS)002024350LBS01-Aleph</subfield>
```

is transformed into:

036
^$ $ $a(BeLVLBS)002024350LBS01-Aleph

- The XML structure:

```
<datafield tag="710" ind1="2" ind2=" ">
        <subfield code="a">Withagius, Joannes I</subfield>
        <subfield code="c">Antwerpen</subfield>
        <subfield code="d">1549-1587</subfield>
        <subfield code="4">bsl</subfield>
```

is transformed into:

710
^2 $aWithagius, Joannes I$ $ $cAntwerpen$ $ $d1549-1587$ $ $4bsl

- The XML structure:

```
<datafield tag="700" ind1="1" ind2=" ">
        <subfield code="a">Ligneus, Petrus</subfield>
```

```
        <subfield code="4">aut</subfield>
    </datafield>
    <datafield tag="700" ind1="1" ind2=" ">
        <subfield code="a">Vergilius Maro, Publius</subfield>
        <subfield code="d">70 B.C.-19 B.C</subfield>
        <subfield code="4">oth</subfield>
    </datafield>
```

Is transformed into:

700
^1 $aLigneus, Petrus$ $ $4aut^$1$ $aVergilius Maro, Publius$ $ $d70 B.C.-19 B.C$ $ $4oth

Tasks to be performed:

1. Standardize and normalize the fields with clustering etc. (you can either isolate the content or work on the original columns: you should decide what is more efficient). When you standardize, never delete the original version, but work on a separate column.
2. Identify fields on which it can be interesting to work and isolate the content from the tags to apply Wikidata reconciliation (for instance people, or places, or titles, or institutions, where available..). If the software cannot search for the proper text in Wikidata, there won't be any match.
3. Be careful not to lose information. If, for instance, you decide to isolate one specific subfield from one column, make sure that the code of the subfield is preserved somewhere (see below for a possible strategy)

450
$1$1$a Some text

can become

450	450**$1$1$a**
$1$1$a Some text	Some text

In the case of Ancient books, additional information (in dutch) can be found here: https://libis.helpdocs.com/datamodellen-documenttypes/oude-drukken