Alephino Conversion control tables

From: Swami Wiki **Date: September 23rd 2014**

Table of Contents

General information

Alephino provides a variety of different formats for data exchange with external systems such as MARC21, MARCXML and ALEPH sequential. In combination with data exchange the following aspects are of importance:

- The formal composition of the respective data format resp. the rules for serialization.
- The character encoding (character set)
- Processing of data contents
- On data import: Rules for merging incoming data with existing records

Data exchange control

The central configuration file etc/alephino.cfg covers subsequentially listed parameters for controlling the Alephino export and import functions. With that predefined settings for known data exchange methods are delivered. These parameters can also be found in input forms of the export and import interface with the Alephino Web services.

Type= Name of the formal type of data:

MARC	MARC21 serial format
ALEPH	ALEPH500 Sequential format
ALEPHINO	Alephino Format
XMARC21	MARC/XML format

ConvIn= Control table for content processing of the input data. The respective file name for the control table is composed of that parameter followed by the 3-digit code of the Alephino master file. i.e. ../etc/marc_alephino**.TIT**

ConvOut= Control table for content processing of the output data. The respective file name for the control table is composed of that parameter followed by the 3-digit code of the Alephino master file. i.e. ../etc/alephino_marc**.TIT**

TranslIn= Table that controls the character encoding translation of the input data. Valid table names are labeled TRANSL= with control file etc/marctransl.int, i.e. **MARCTOEXT**.

TranslOut= Table that controls the character encoding translation of the output data. Valid table names are labeled TRANSL= with control file etc/marctransl.int, i.e. **EXTTOMARC**.

Generic=Y/N The formatting of a record for export is controlled by its internal definition, by which its general structure, fields (tags), sub-fields, repeatablility of tags and indicators are set.

On export formatting for different bibliographic schema, i.e. conversion from german standard MAB2 into MARC21 or vice versa the above mentioned method is not applicable. Instead a generic output formatting method is used.

ScriptIn= Import processing by Alephino script. For parameter the name of a script made available by having it labelled *SCRIPT*= in the Alephino control tables is to be used. The script processing is performed after initial data conversion controlled by conversion table defined by parameter *ConvIn*=

ScriptOut= Export processing by Alephino script (see above). The script processing is performed after data conversion controlled by conversion table defined by parameter *ConvOut*=

PSIIn= Import processing by PSI script. For parameter the file name (path name) of a PSI script is to be used. The PSI script processing is performed after data conversion and script processing controlled by parameters ConvIn= and ScriptIn=.

PSIOut= Export processing by PSI script. For parameter the file name (path name) of a PSI script is to be used. The PSI script processing is performed after data conversion and script processing controlled by parameters ConvOut= and ScriptOut=.

XSLT= On import of an XML file, an XSL stylesheet can be specified, which realizes the pre-processing or conversion in a MARC/XML format. The stylesheet is to be located in the directory **etc/xslt**; whose file name is specified without a leading path and without extension ".xsl".

Conversion control tables

For description of contents conversion processing while data exchange with external systems the following format is used:

- Every line describes the conversion rule for a certain field resp. subfield.
- An asterisk ***** at the beginning of a line makes a comment.
- **DEFAULT** at the beginning of a line defines the conversion rule for a source field that lacks a special rule. If no procedure name is given subsequentially, fields without a specific rule will be discarded. If no DEFAULT line exists fields without specific rule will be passed through without modifications.
- A line may consist of 4 columns separated by **Tab character**:

Source
Source field in fully qualified Alephino notation (whereas sub-field, repetition factor and indicators can be declared).

Target
Target field in fully qualified Alephino notation (whereas sub-field, repetition factor and indicators can be declared).

Program (optional, max. 29 digits)
Name of a conversion routine to be used for content preparation of that field as long as no simple copying is desired.

Parameter (optional, max. 19 digits)
Additional data depending on the conversion routine. If blank characters are contained, the entire column must be enclosed in double quotes.

Programs (routine names) for content preparation of Alephino data fields

Name	Function	Parameter
cnv_norm	Pass through including all repetition a field may have. This function is used implicitely if the table lacks a DEFAULT declaration.	
cnv_copy	Copy field to output.	
cnv_increm	Increase tag number of the output field beginning from declared field.	n = step width
cnv_update	Insert field (if not already exists), otherwise update.	
cnv_cat	Append content to output field.	Use separator string if output field is already filled.
cnv_delete	Delete field (do not pass to output).	
cnv_prefix	Copy field with leading string constant.	Prefix
cnv_suffix	Copy field with trailing string constant.	
cnv_all_subfields	Concatenate all sub-fields of the source field using **/** for separator to target field.	
cnv_arrange_subfd	Concatenate all sub-fields ordered by parameter list using separator from parameter list to target field.	Subfield order declaration string. As an option the separator string can be declared with parameter enclosed in **|**.
cnv_cond_tag	Conditional conversion - will be performed if the field declared as parameter does exist.	
cnv_cond_notag	Conditional conversion - will be performed if the field declared as parameter does not exist.	
cnv_tabkey	Table driven contents conversion.	Name of the table labeled FORM=TABLE with the Control tables.
cnv_substr	Partial string conversion	Initial position, length

Conversion functions in cataloging module

Deviating from Aleph the Alephino does not use the "hard-coded" conversion methods available with the GUI client but rather the **REMOTE** method. With that all sorts of data conversions will be performed at server side. At the moment the following conversion methods are available by default with the cataloging module:

catalog/tab/catalog.ini:

```
[ConvertFile]
Convert1=L,MARC21 Titles (MARC-ISO MARC8),REMOTE,MRCTITISOMRC
Convert2=L,MARC21 Titles (XML UTF-8),REMOTE,MRCTITXMLUTF
```

The emphasized codes at the end of the above lines are references to conversion procedures defined at the server in file **etc/catconv.cfg**:

```
* MARC Title (Charset MARC8)
MRCTITISOMRC,TYPE=MARC21,TRANS=MARCTOEXT,HEAD=245$a
* MARC/XML Title (Charset UTF-8)
MRCTITXMLUTF,TYPE=XMARC21,HEAD=245$a,XSLT=marcxml
```

The parameter TYPE, TAB, TRANS and XSLT are the same as described in the above chaptern for Type, ConvIn, TranslIn and XSLT, while HEAD means the field that shall be used in short list format for preview of converted records.