Working with MARC Tools

TERRY REESE

GRAY FAMILY CHAIR FOR INNOVATIVE LIBRARY SERVICES

TERRY.REESE@OREGONSTATE.EDU

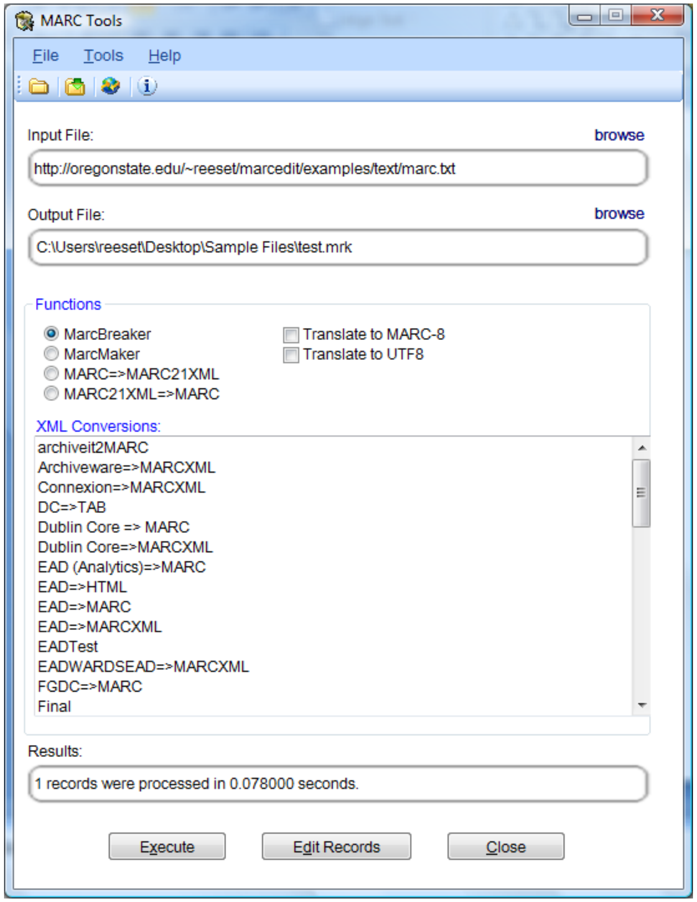
Roadmap

- What is the MARC Tools section
- MarcBreaker
- MarcMaker
- Charactersets
- XML Functions
  - How does it work
  - Building/Editing new XML Functions
- MARCSplit/MARCJoin

MARC Tools

- What is the MARC Tools section
  - Access to the Making and Breaking functionality
  - Characterset processing
  - Access to the XML Sub-routines

MARC Tools



Input File: browse

http://oregonstate.edu/~reeset/marcedit/examples/text/marc.txt

Output File: browse

C:\Users\reeset\Desktop\Sample Files\test.mrk

Functions

- ● MarcBreaker
- ○ MarcMaker
- ○ MARC=>MARC21XML
- ○ MARC21XML=>MARC
- □ Translate to MARC-8
- □ Translate to UTF8

XML Conversions:

archiveit2MARC
Archiveware=>MARCXML
Connexion=>MARCXML
DC=>TAB
Dublin Core => MARC
Dublin Core=>MARCXML
EAD (Analytics)=>MARC
EAD=>HTML
EAD=>MARC
EAD=>MARCXML
EADTest
EADWARDSEAD=>MARCXML
FGDC=>MARC
Final

Results:

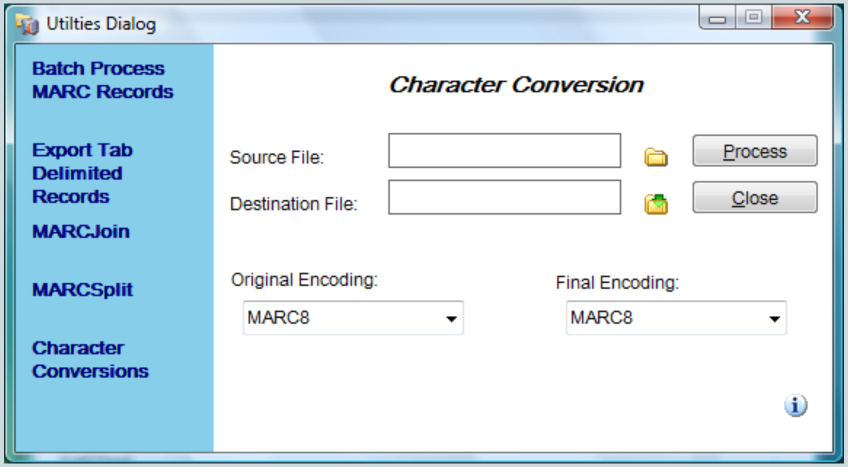
1 records were processed in 0.078000 seconds.

Marc Tools

- Built-in functions
  - MarcBreaker – Tool used to convert MARC records to the MarcEdit mnemonic format
  - MarcMaker – Tool used to convert MarcEdit mnemonic format to MARC
  - MARC=>MARC21XML – converts MARC to MARC21XML
    - Automatically converts data from MARC-8 to UTF8
  - MARC21XML=>MARC – converts MARC21XML to MARC
    - Doesn't automatically convert data from UTF8 to MARC8 – will leave data in UTF8

MARC Character Conversions

- Supports moving between any known Windows Characterset and MARC8.
- Can be run from the Breaker/Maker – or as its own standalone utility

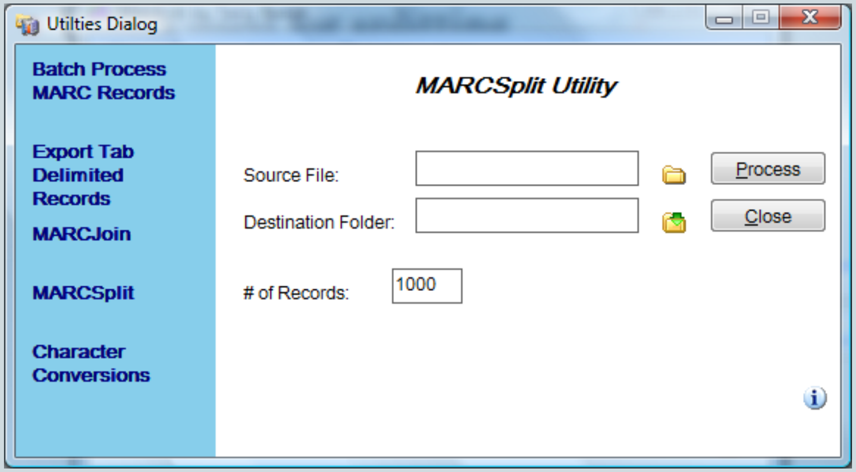


Original Encoding: MARC8

Final Encoding: MARC8

MARCSplit

- Utility used for splitting large MARC record sets into smaller files



Utilties Dialog

MARCSplit Utility

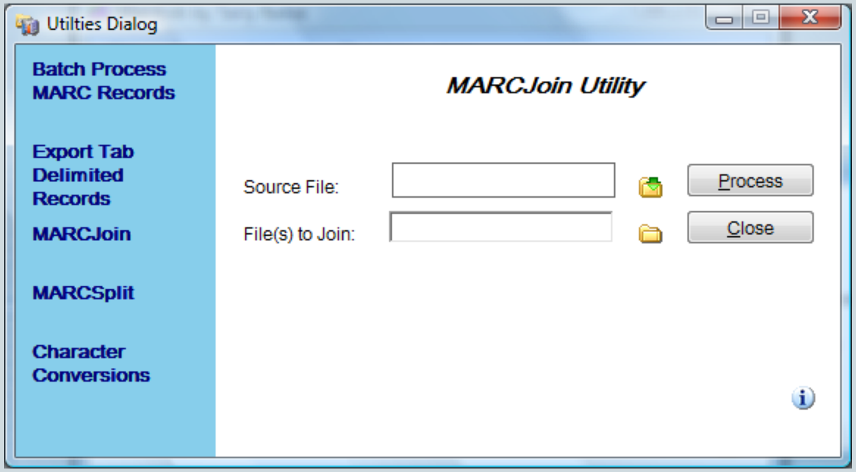
Source File:

Destination Folder:

# of Records: 1000

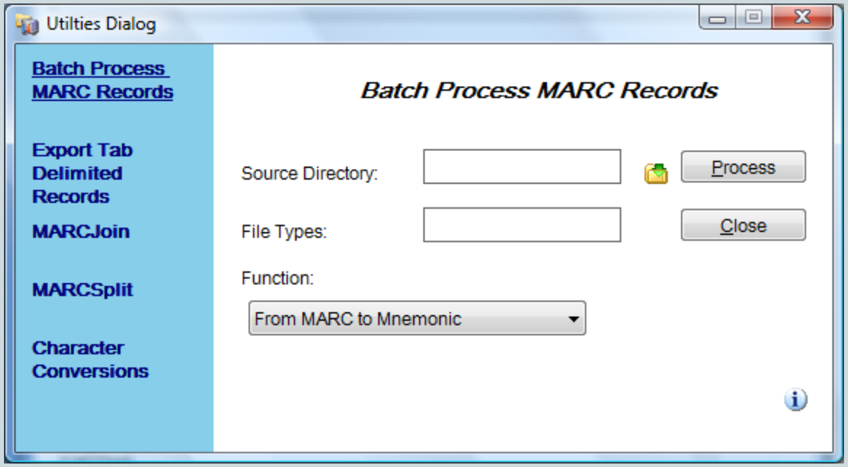
MARCJoin

- Utility used for joining large sets of MARC data to a single file



Batch Record Processor

- Allows MarcEdit to process “lots” of files.
- Files can be processed against an entire folder’s contents or by file type
- Can utilize any built-in or derived XML Function transformation



Utilties Dialog

Batch Process MARC Records

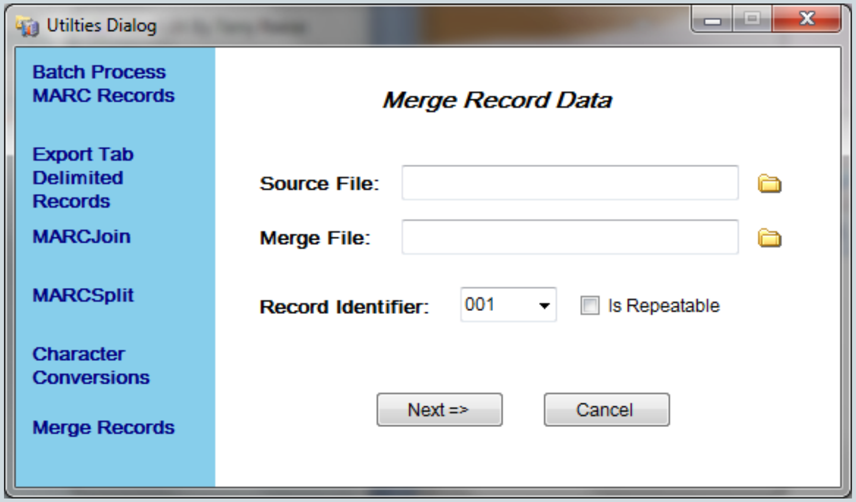
Source Directory:

File Types:

Function: From MARC to Mnemonic

Merge Records Tool

- Allows users to merge MARC data from two files
- Allows users to merge unique data, selected data and all data.



Merge Record Data

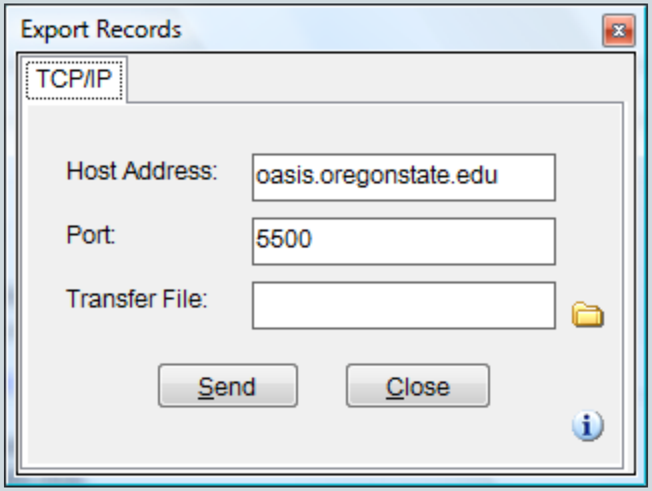
Source File:

Merge File:

Record Identifier: 001 ☐ Is Repeatable

Exporting Data

- Exports raw MARC data using TCP/IP connections
- Can export individual or multiple records (regardless of ILS system support for batch record import)
- If set in the preferences, will capture any data returned back from the server



Export Records

TCP/IP

Host Address: oasis.oregonstate.edu

Port: 5500

Transfer File:

Little Known Functionality

- Ability to process remote records from within the MARC Tools area
  - Can be either MARC or XML data
  - Can be zipped (if zipped, the file will be preprocessed)

Little Known Functionality

- XML Processing
  - MarcEdit's MARC Tools supports multiple XML engines, settings.
    - MSXML (Microsoft .NET optimized XSLT engine)
    - Saxon.NET (.NET version of Saxon)
  - Why do we care?
    - MSXML processor
      - Small and fast, Optimized for low memory footprint and speed and Supports only XSLT 1.0/Xquery 1.0
    - Saxon.NET
      - Supports XSLT 2.0, Xquery 2.0, Little slower, but more flexible and allows use of Saxon XSLT extension when writing XSLT documents.

Little Known Functionality

- Character conversion isn't limited to known – pre-populated items. You can define your own character-sets for process.
  - MarcEdit can utilize any known Windows Characterset when doing characterset conversion.
    - Defined MS charactersets: http://msdn.microsoft.com/en-us/library/dd317756(VS.85).aspx

MARC Tools Demo

- MarcBreaker
  - Local file and remote
  - Invalid file
- MarcMaker
- Character conversion
- MARCSplit/MARCJoin

MARC Conversions

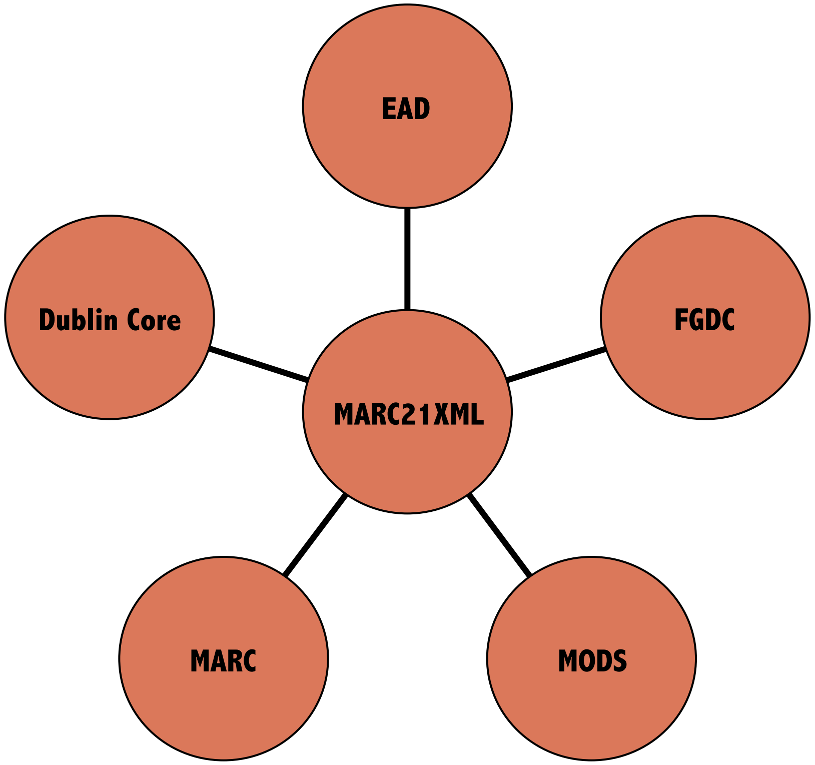
MarcEdit: crosswalking design

- MarcEdit model:
  - So long as a schema has been mapped to MARCXML, any metadata combination could be utilized. This means that no more than two tranformations will ever take place. Example: MODS → MARCXML → EAD

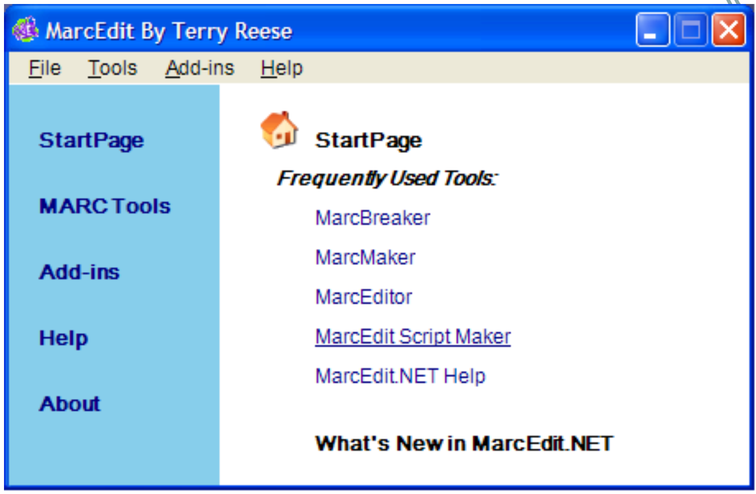
MarcEdit: crosswalking design

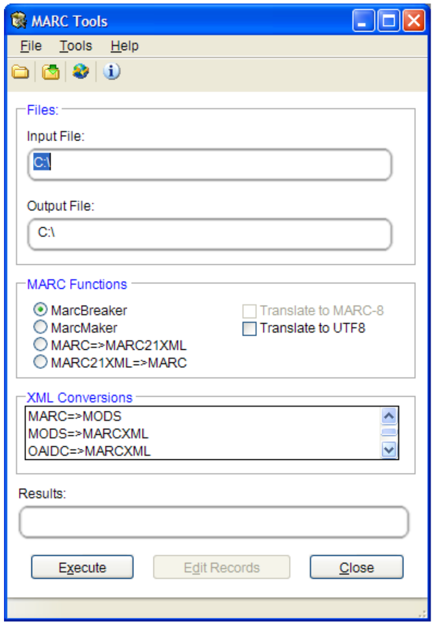
- MarcEdit Crosswalk model
  - Pro
    - Crosswalks need not be directly related to each other
    - Requires crosswalker to know specific knowledge of only one schema
  - Con
    - each known crosswalk must be mapped to MARCXML.

MarcEdit Crosswalking model



MarcEdit: Crosswalks for everyone





Files:

Input File: C:\

Output File: C:\

MARC Functions

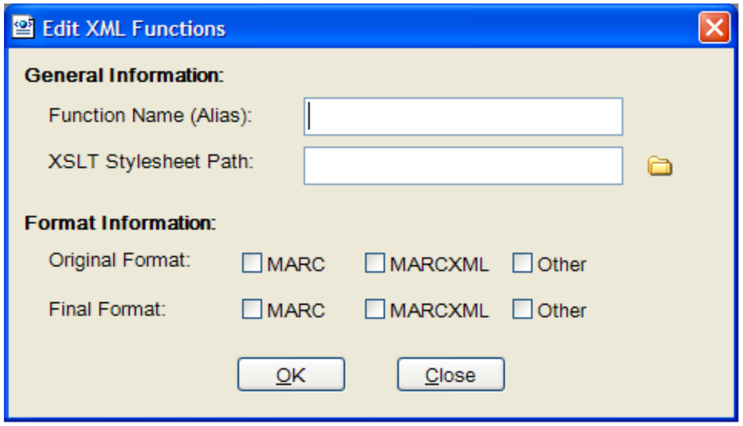
● MarcBreaker
○ MarcMaker
○ MARC=>MARC21XML
○ MARC21XML=>MARC

□ Translate to MARC-8
□ Translate to UTF8

XML Conversions

MARC=>MODS
MODS=>MARCXML
OAIDC=>MARCXML

Results:



General Information:

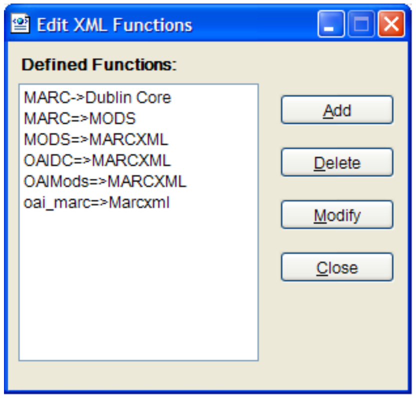
Function Name (Alias):

XSLT Stylesheet Path:

Format Information:

Original Format: □ MARC □ MARCXML □ Other

Final Format: □ MARC □ MARCXML □ Other



Defined Functions:

MARC->Dublin Core
MARC=>MODS
MODS=>MARCXML
OAIDC=>MARCXML
OAIMods=>MARCXML
oai_marc=>Marcxml

MarcEdit: Crosswalks for everyone

- Example Crosswalks:
  - MODS => MARC
  - MODS => FGDC
  - MODS => Dublin Core
  - EAD => MODS
  - EAD=>HTML

MarcEdit: Crosswalks for everyone

- What's MarcEdit doing?
  - Facilitates the crosswalk by:
    1. Performing character translations (MARC8-UTF8)
    2. Facilitates interaction between binary and XML formats.

Examples

- Project Gutenburg RDF => MARC
- EAD=>MARC