Introduction to MarcEdit

Workshop

R. Antonio Muñoz Gómez, Metadata librarian
Sarah Gorman, TALint student - Jordan Pedersen, TALint student

Learning objectives

By the end of this workshop, participants will be able to:

1. Understand some of the general capabilities of MarcEdit
2. Use MarcEdit to perform simple tasks, such as:
    a. Obtain bibliographic records using the Z39.50 client
    b. Apply batch transformations to bibliographic records, like adding, deleting, and/or modifying fields, subfields, and indicators
3. Understand basic principles of regular expressions (regex) and their usefulness for data cleanup

Preliminary notes

MarcEdit is free software programmed by Terry Reese, librarian at the Ohio State University.
You may download a copy from: http://marcedit.reeset.net/
In this workshop, we will use version 6.3 of the software, which was the latest stable version at the time that this workshop and handout were put together. Since then, a new version (MarcEdit 7) has been released. The skills that you will learn in this workshop are perfectly compatible with the newest version of the software.

Introduction

MarcEdit : Main Interface Options

Tools we use:

MARC Tools

1. **MarcBreaker** – Tool used to convert MARC records to the MarcEdit mnemonic format (see “Compile File into MARC” for more info).
2. **MarcMaker –** Tool used to convert MarcEdit mnemonic format to MARC
3. Converting between MARC formats [i.e. MARC=>MARC21XML (and vice versa)]
4. Converting between different metadata schemas: [i.e. JSON=>MARC (and vice versa), Dublin Core=>MARC(and vice versa)]

Delimited Text Translator

Map fields in a delimited text file into a MARCXML or MARC file. Can be used to create MARC records from excel documents.

MarcEditor

A metadata editor- this is what we mainly work with, and what you will learn in the workshop today.

Z39.50/SRU Client

Allows users to query databases using the z39.50 protocol and save the records in MARC form or browse them in a MARC notepad format (similar to the Mnemonic .mrk format), make changes to the record and download it. Library databases that are drawn from must be pre-configured, which will be covered in the workshop.

Other tools:

Harvest OAI Records

Harvest records from collections which offer OAI-PMH (Open Access Initiative Protocol for Metadata Harvesting) services in a variety of metadata schemas. Can bring in individual records, records from a specific date, or large batches of records from set lists.

MARCNext

Proof of concept tools. Originally created with the intention of allowing librarians to take existing metadata and model it using Bibframe, it now includes other tools such as a SPARQL Browser, Link Identifiers and OpenRefine Integration.

MARC SQL Explorer

Allows users to export their data into either an SQLite or MySQL data and run SQL queries directly on the data to mine for information.

Configuring the Z39.50 Client

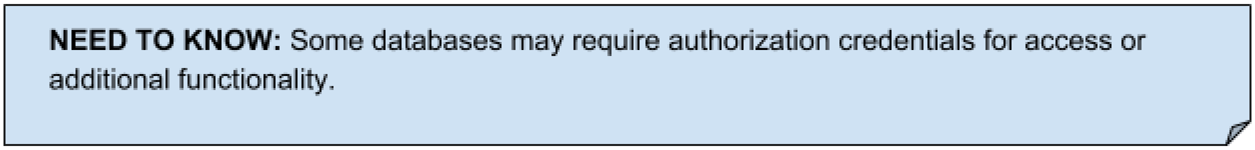
MarcEdit Z39.50 client In order to obtain records from other institutions, we first need to configure our Z39.50 client, adding the database(s) that we want to query. The Z39.50 Target Directory (http://irspy.indexdata.com/) has the names, addresses, ports, and other technical information required to query library databases and obtain MARC records.

Step 1 : Find a database to query. Go to the Z39.50 Target Directory (http://irspy.indexdata.com/) and use the Find a target option to find a library whose database you would like to query. Some examples include: the University of Toronto, University of Alberta, Toronto Public Library.

Step 2 : In MarcEdit > Z39.50/SRU Client > Modify Databases > Add Database

Step 3 : Copy-paste the relevant information in the corresponding fields: • Name • Host • Database • Port • Syntax
Step 4 : Check Show Database
Step 5 : Save
For example:

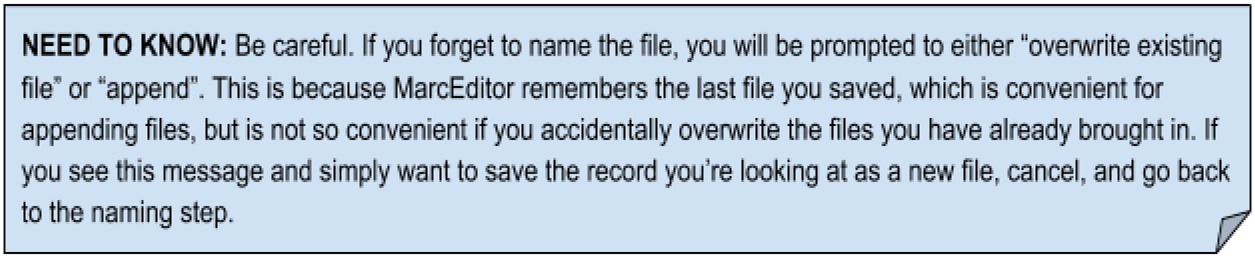
- Name: Toronto Public Library
- Host: symphony.torontopubliclibrary.ca
- Database: unicorn
- Port: 2200
- Syntax: usmarc



NEED TO KNOW: Some databases may require authorization credentials for access or additional functionality.

Obtain records (OCLC/local records/others)

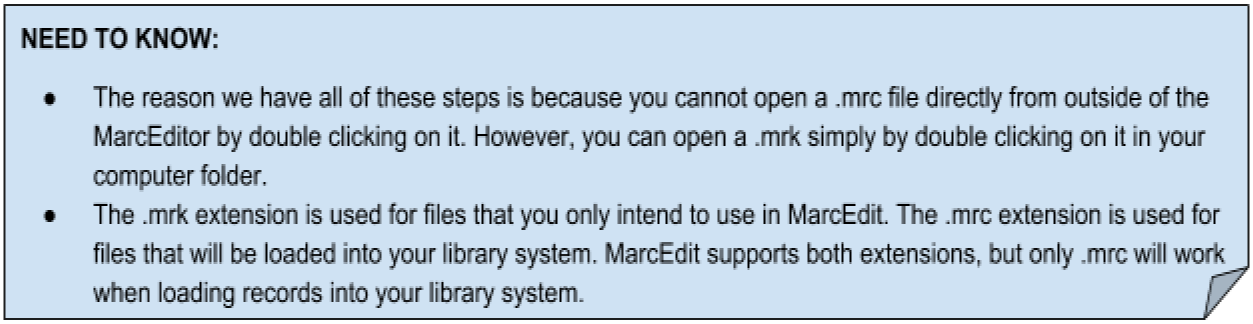
1. Open Z39.50/SRU (double click in the main interface)
2. Click on Select Database in the options menu to choose the database you want to bring the MARC records from.
3. Ensure “Retrieve Unicode is checked, and Server Character set is listed as UTF8
4. Paste ISBN, record number, title etc. into the Search box and ensure you have the corresponding type chosen in the dropdown menu. We recommend using record number if possible.
5. Click the green arrow to search.
6. Double click on the record to open it.
7. You will be able to view the file, if it is what you want, click on the yellow folder at the top of the menu.
8. Type in the name with which you want to save the document and click save.
9. But wait! The file is not actually saved yet. At this point you can choose to download the record, and you will have a file with one record in it. However, if you have already saved a record, and you want to have a file with multiple records in it, click the “Append” box beside the yellow folder before hitting download. Once you click download, the record will be saved in the location you selected



NEED TO KNOW: Be careful. If you forget to name the file, you will be prompted to either “overwrite existing file” or “append”. This is because MarcEditor remembers the last file you saved, which is convenient for appending files, but is not so convenient if you accidentally overwrite the files you have already brought in. If you see this message and simply want to save the record you’re looking at as a new file, cancel, and go back to the naming step.

Opening Files (ctrl+o)

- Open MarcEditor (double click in the main interface).
- Click on File menu in the top left corner of the window, and then Open.
- On the bottom right hand drop down menu, ensure you are looking for .mrc files (see note below).
- Double click on the file to open.



NEED TO KNOW:

- The reason we have all of these steps is because you cannot open a .mrc file directly from outside of the MarcEditor by double clicking on it. However, you can open a .mrk simply by double clicking on it in your computer folder.
- The .mrk extension is used for files that you only intend to use in MarcEdit. The .mrc extension is used for files that will be loaded into your library system. MarcEdit supports both extensions, but only .mrc will work when loading records into your library system.

General workflow and best practices

Saving Files and File Copies

MARCEdit files can be saved in two ways – using “Save As…” or using “Compile File into MARC.”

Save As…:

- The “Save as” option saves your file as a **.mrk** file. These files can be saved with a shortcut key.
- When using “save as,” change the file type to “**MARC UTF-8 Text File**” in the save menu. This will ensure that characters display correctly in the file.

Compile File into MARC:

- The “Compile File into MARC” option saves your file as a **.mrc** file. These files must be re-compiled each time you want to save them.
- When compiling a file into MARC, change the file type to “**UTF-8 MARC File**” in the save menu. This will ensure that characters display correctly in the file and in your library catalogue.

Saving File Copies

- You should *always* save a file copy (either in .mrc or .mrk) before you start working and before you make any edits you are unsure about. Some edits can change your file irreversibly.

<u>Undo and Special Undo</u>

Undo (F2):

- Will undo the most recent *manual edits* one at a time.
- Manual edits are any changes you make by directly clicking and typing into a record, just like you would in a normal document.

Special undo:

- Will undo the most recent *batch edit*. Will also undo any manual edits that you made since that batch edit.
- Batch edits are any changes you make using the MARCEdit “Tools” menu. We will talk about these later.

Key things to know:

- You only get *one* special undo!
- Once you make a batch edit, nothing changed before that batch edit can be undone!

<u>Field Count Reports</u>

Field Count Reports can be used to get a sense of what’s in your file. These reports will show how many times each individual field appears in your file, which is especially helpful if you are dealing with hundreds or thousands of records at once.

- To count fields, choose the “Reports” menu, then “Field Count”
- A field count report gives two kinds of information: The total occurrences of each field and the number of records that contain the field at least one time.
  - If a field occurs 3 times and it is found in 1 record, this means that field is repeated 3 times within that 1 record.
- Field 000 (at the very beginning) indicates the total number of records.

Key things to know:

- You should always run field counts before making any batch edits and to record the number of occurrences of each field you plan to edit. This lets you compare the changes you expect with the changes you actually make.

Field Overviews

It is also possible to display every occurrence of one single field in your file. This lets you compare fields at a glance to get a sense of the data available, changes needed, and potential errors that may arise/need troubleshooting.

- To display occurrences of a field, choose the “Edit” menu, then “Find” *or* press Ctrl+F
- Type your field number, preceded by the ‘=’ sign, then select “Find All”

- Once you see your field overview, you can select a particular occurrence of the field and choose “Jump to Page” to go to that occurrence in your file and modify the record or get a closer look at the problem.

Key things to know:

- Use field count reports to record information about fields before batch editing.
- Use field overviews to double-check your work, diagnose problems, and navigate to small issues in need of manual editing.

Changing Display Settings

Before we move on... you can change the way that your records display in MARCEdit by choosing Tools, then Preferences, then at the left of the menu that pops up, choose MarcEditor and you will see options for changing the font and text size.

You can also automatically set records to compile as UTF8 so you don’t have to select this option every time you compile your file.

MarcEdit Notation

In MarcEdit, fields, subfields, indicators, and blank spaces are written slightly different from what you may be used to in Sirsi.

	MarcEdit	**Sirsi**
Fields	=245	245
Subfields	$	|
Blank spaces (e.g. indicators)	\	[nothing]

Example:

MarcEdit	=300 \\$a40 pages :$billustrations ;$c28 cm.
Sirsi	300 _ _ |a40 pages :|billustrations ;|c28 cm.

NOTE: In MarcEdit, there are two spaces between the field number and the indicators.

Practical applications: MarcEditor tool

For this part of the workshop, we will be using the file called **workshopRecords002.mrk8**

Add / delete fields

Exercise 1: Add field 039 with your initials in subfield $f

STEP 1 : Prepare

```
BEFORE:             =008  750902s1904\\\\nyu\\\\\\\\\\\000\0\eng\d
                    =040  \\$aDLC$beng$cHUL

AFTER:              =008  750902s1904\\\\nyu\\\\\\\\\\\000\0\eng\d
                    =039  \\$framg
                    =040  \\$aDLC$beng$cHUL
```

Field count
In MarcEditor > Reports > Field count

Field 039: _____ occurrences. Found in _____ records

STEP 2 : Transform

In MarcEditor > Tools > Add/Delete Field
OR
Keyboard shortcut: F7

Field: 039
Field Data: \\$framg

Options (if applicable):
Check: Add field only if not present
Check: Add field only if not a duplicate

Add Field

NOTE: When adding a new field, you MUST ALSO include indicators AND subfields as part of the Field Data.

STEP 3 : Quality control

In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =039
Find All

Check that the work you have done meets your expectations. If not, use the undo or special undo functions as necessary, and then repeat or modify the process until you get the desired result.

STEP 4 : Save your progress

In MarcEdit > File > Save

OR

Keyboard shortcut: Ctrl+S

OR

Click on the save icon

Exercise 2: Delete field 598 \\$aNEW

STEP 1 : Prepare

BEFORE: =598 \\$aNEW
AFTER: [No 598 field]

Field count
In MarcEditor > Reports > Field count

Field 598:   _____ occurrences. Found in _____ records

Examine the information currently present in field 598 to confirm that the field can be safely deleted.
In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =598

Find All

STEP 2 : Transform

In MarcEditor > Tools > Add/Delete Field
OR
Keyboard shortcut: F7

Field: 598

Delete Field

NOTE: IF you had some fields 598 with other information that you wanted to keep, you could limit your transformation by adding details in the Field Data

Example: Field: 598
Field Data: $aNEW
Delete Field

This option would ONLY delete 598 IF there is $aNEW in it.

STEP 3 : Quality control

In MarcEditor > Reports > Field count

Field 598: _____ occurrences. Found in _____ records

The report should have NO 598 fields.

STEP 4 : Save your progress

In MarcEdit > File > Save

OR

Keyboard shortcut: Ctrl+S

OR

Click on the save icon 💾

Edit Indicator Data

Exercise 3: Fix second indicator in field 245

STEP 1 : Prepare

BEFORE: =245 10$aThe Little Mermaid
AFTER: =245 1**4**$aThe Little Mermaid

In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =245 10$aThe
Find All

NOTE: In your search AND in your transformation, you MUST include a space after article The. If you don't, you may inadvertently change the indicators for 245 fields whose initial words start with The (Theology, Thesaurus, etc).

STEP 2 : Transform

In MarcEditor > Tools > Edit Indicator Data
OR
Keyboard shortcut: F8

Field:	245
Indicators:	10
Field Data:	$aThe
Replace with indicators:	14

Replace

NOTE: You may have noticed that the number of modifications made will be larger than the number of 245 fields in your file. This is because each field has two indicators, so any transformation made to the indicators will be double the number of corresponding fields.

STEP 3 : Quality control

In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =245
Find All

Check that the work you have done meets your expectations. If not, use the undo or special undo functions as necessary, and then repeat or modify the process until you get the desired result.

STEP 4 : Save your progress

Working with variable data: Basics of regular expressions

Sometimes we want to make transformations that affect multiple similar fields, each of which has unique data.

Example:

Field 020 normally contains the ISBN for the book in subfield $a. It may also contain qualifying information in subfield $q, such as whether the ISBN applies to the paperback edition or hardcover.

```
=020  \\$a0802142176$qpbk.
```

Subfield $q was introduced in 2013. Prior to that date, qualifying information was contained in subfield $a. It is possible to find 020 fields which look like this:

```
=020  \\$a9780802078285 (pbk.)
```

We would like to fix all 020 fields in our file, so that qualifying information is moved to subfield $q.

However there are a few challenges:

1. The length of the ISBN is variable (10 or 13 digits).
2. The qualifying information is also variable:
    a. It may or may not include parentheses.
    b. It may appear abbreviated or in full (e.g. paperback OR pbk.)
    c. Different words or abbreviations may appear as qualifying information (e.g. hardcover, pbk., vol.1)

To address this type of issue, we use regular expressions. Regular expressions identify a general pattern in simplified form, that applies to all the text strings that we want to transform, and then help us find and replace any strings that fulfill our criteria.

Exercise 4: Identifying a pattern

What do these 020 fields all have in common?

```
=020 \\$a9780802008435 (bound)
=020 \\$a9780802078285 (pbk.)
```

```
=020 \\$a9781895830835 (Paper)
=020 \\$a0195057317 (v. 1)
```

Solution :

1. All of them have a subfield $a
2. All of them have some number of digits from 0-9
3. All of them have a space after the digit string
4. All of them are followed by an opening parenthesis
5. All of them have letters following the parenthesis

With the above information, let us begin writing our regular expression:

```
$a      0-9     space  (        letters
```

In regular expressions, certain symbols have special meanings, such as $ and (. In order for our regular expression to work, we need to “escape” these characters if we want them to keep their literal value. We use the backslash (\) to escape characters:

```
\$a     0-9     space  \(       letters
```

The space is represented by \s:

```
\$a     0-9     \s     \(       letters
```

Our regular expression includes a range, in this case, of digits between 0 and 9. We indicate a range using square brackets [ ]:

```
\$a     [0-9]   \s     \(       letters
```

We don’t know exactly how many digits are in our string, it could be 10, it could be 13 (it could be more if there was a typo when entering this information). We tell the computer to look for this range of numbers appearing “any number of times” using the + symbol:

```
\$a     [0-9]+  \s     \(       letters
```

NOTE: We can also use the * symbol to indicate “any number of times”. The difference is that * assumes “any number of times, including 0 times”, whereas + assumes “any number of times, at least 1”.

We don’t know exactly which letters appear after the parenthesis, nor do we know if they are uppercase or lower case. We have two options.

Option 1:
Use ranges

[a-z,A-Z]*

Option 2:
Use a period to indicate “any character whatsoever, including spaces, digits, letters, but excluding line breaks”.

.*

Our regular expression will look something like this:

```
\$a[0-9]+\s\(.*
```

Remember that we want to add a subfield $q instead of the blank space. We can do this by grouping parts of our regular expression that we want to keep unchanged, and separate groups for parts we want to change. We use parentheses to group parts of our regular expression:

(\$a[0-9]+)	:	We want to keep subfield $a, as well as any number of digits in it.
(\s)	:	We want to group the blank space by itself, because we will be changing it.
(\(.*)	:	We want to keep the opening parenthesis, as well as any characters after it.

Now our regular expression will look like this:

```
(\$a[0-9]+)(\s)(\(.*)
```

We can identify each part of our regular expression using a number starting with 1:

(\$a[0-9]+)	(\s)	(\(.*)
1	2	3

Now we will tell the program to keep part 1, replace part 2 with $q and keep part 3. Here is how we do it:

```
$1$q$3
```

Now let’s put everything together.

STEP 1 : Prepare

BEFORE: =020 \\$a9780802078285 (pbk.)
AFTER: =020 \\$a9780802078285**$q**(pbk.)

In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =020.*(\$a[0-9]+)(\s)(\(.*)
Find All

STEP 2 : Transform

In MarcEditor > Edit field data
Field: 020
Find: (\$a[0-9]+)(\s)(\(.*)
Replace: $1$q$3
Options: Check “Use Regular Expressions”

Process

STEP 3 : Quality control

In MarcEditor > Edit > Find
OR
Keyboard shortcut: Ctrl+F

Find what: =020
Find All

Check that the work you have done meets your expectations. If not, use the undo or special undo functions as necessary, and then repeat or modify the process until you get the desired result.

STEP 4 : Save your progress

RDA Helper

In MarcEditor > Tools > RDA Helper

CAUTION:

- MarcEdit will add 336, 337, and 338 based on information in the LDR field. Coding mistakes in the LDR may result in the wrong RDA fields.
- Even correctly coded LDR fields may result in the wrong RDA fields if the algorithm cannot programatically infer the right content, media, and/or carrier types.
- It is best to use the RDA Helper with sets of records that are very homogeneous in terms of content, media, and carrier types.
- “Safe” uses of the RDA helper include expanding abbreviations, deleting GMD, and adding $erda to field 040.

Task lists

In MarcEditor > Tools > Manage Tasks > Create New Task List

1. Enter the name for your task list and click OK
2. Select the newly-created task list from the menu
3. Click on “Edit Selected Task List”

You can now add common steps that you would use regularly for your record cleanups.

CAUTION:

Task lists will run all the specified changes to all the records in the file. They work well when your record-set is very homogeneous and the changes you need to make are fairly consistent. It is not advisable to run a task list on a file that has mixed records which may require specific types of changes (e.g. if your file has a mix of print and electronic records).

Learn more

MarcEdit

http://marcedit.reeset.net/help
RECOMMENDED: Join the MarcEdit Listserv!

Documentation from the University of Illinois at Urbana-Champaign
http://guides.library.illinois.edu/c.php?g=463460&p=3167871

Regular expressions

https://regexr.com/