

Creating Radioactive MARC Records and Z Queries Using the MARCdocs Database

Draft – December 2, 2004

Prepared by
William E. Moen

Contents

Introduction	1
The MARCdocs Database	1
Requirements for Extensions to MARCdocs Database	2
Proposed Extensions to MARCdocs Database	3
Token Requirements to Support Profile Level 0 and 1 Searches for Author, Title, Subject	4
Format of the Tokens	4
Using the Extended MARCdocs Data to Support RadMARC Record Creation	7
Using the Extended MARCdocs Data to Support Z-Query Creation	10
Summary and Conclusion	10

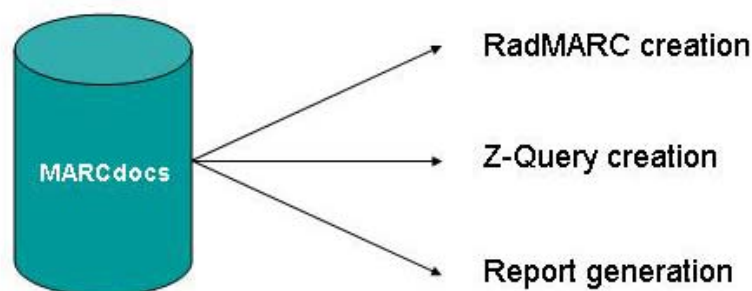
Introduction

This document describes how we can extend a relational database of MARC documentation to store the appropriate information that will support the automatic generation of the special, diagnostic MARC records we call radioactive MARC (RadMARC) records. The information contained in the database will also support the generation of the Z queries used in the interoperability testing.

The MARCdocs Database

Over the summer and early fall, project staff created an application to store information from the *MARC 21 Concise Format for Bibliographic Data*. MARCdocs, the MARC 21 Documentation Database, is a pilot effort aimed at structuring the textual documentation from the MARC 21 Format for Bibliographic Data into a relational database. Using a database approach for the authoritative MARC documentation provides new opportunities for various applications. This application uses open source software tools including Linux, MySQL, and PHP.

Based on a suggestion from Sebastian Hammer, Index Data, the project staff has discussed how the MARCdocs application could be extended to support higher levels of automated MARC record creation, interoperability testing, and report generation, as indicated in the figure below.



Requirements for Extensions to MARCdocs Database

Two primary pieces of information need to be stored in the MARCdocs database to support the RadMARC and Z-Query creation:

- Indication of which subfields could be indexed to support U.S. National Profile searches, and which specific searches are supported
- One or more tokens for each of these subfields

From previous research, we know that the occurrences of MARC content designation vary widely. We did a frequency count of MARC content designation used in the Z-Interop test dataset of more than 400,000 MARC records from OCLC's WorldCat database. The table below shows a sample of the frequency analysis.

MARC 21 Field	MARC Subfield	Occurrence
001		419,657
003		419,657
005		419,657
006		652
007		30,556
008		419,657
010	a	305,407
010	b	2
010	z	6,627
650	2	15,361
650	6	9
650	a	602,362
650	b	28
650	c	4
650	k	2
650	v	83,607
650	x	326,867
650	y	32,728
650	z	231,459

Further, we analyzed the occurrence of content designation that could be indexed to support Author, Title, and Subject searches, and discovered that 19 of the more than 500 subfields that could be indexed accounted for 80% of all occurrences. The table below shows these 19 subfields.

# of Occurrences	Marc 21 Field	Subfield	Description	Index
602,362	650	a	Subject added entry Topical Term Subfield a = Topical term or geographic name as entry element	Subject
419,641	245	a	Title Statement Subfield a = Title	Title
329,796	245	c	Title Statement Subfield c = statement of responsibility	Author
326,867	650	x	Subject added entry Topical Term Subfield x = General subdivision	Subject
318,692	100	a	Main entry Personal Name Subfield a = personal name	Author
231,459	650	z	Subject added entry Topical Term Subfield z = Geographic subdivision	Subject
176,916	700	a	Added entry Personal Name Subfield a = personal name	Author
169,178	245	b	Title Statement Subfield b = Remainder of title	Title
149,540	100	d	Main entry Personal Name Subfield d = dates associated with a name	Author
118,647	651	x	Subject added entry Geographic Name	Subject

# of Occurrences	Marc 21 Field	Subfield	Description	Index
			Subfield x = General subdivision	
113,050	651	a	Subject added entry Geographic Name Subfield a = Geographic name	Subject
83,607	650	v	Subject added entry Topical Term Subfield v = Form subdivision	Subject
74,606	700	d	Added entry Personal Name Subfield d = dates associated with a name	Author
69,636	600	a	Subject added entry Personal Name Subfield a = personal name	Subject
66,375	710	a	Added entry Corporate Name Subfield a = corporate name or jurisdiction name	Author
64,433	440	a	Series Statement Added Entry Title Subfield a = title	Title
62,853	490	a	Series Statement Subfield a = Series statement	Title
56,229	600	d	Subject added entry Personal Name Subfield d = dates associated with a name	Subject
55,311	653	a	Index Term Uncontrolled Subfield a = the term	Subject

We can identify various sets of subfields that could be indexed to support the U.S. National Profile Level 0 searches, and use the frequency count from the previous analysis to select sets of subfields. The selection could be based on a threshold of occurrence (e.g., select all subfields that could be indexed to support the Author Keyword Search that occur more than 100,000 times). This would yield a smaller number of subfields per record that needed to be populated.

A third piece of information that needs to be stored in the MARCdocs database is the frequency count for all content designation. The frequency count number will be taken from our previous analysis.

Proposed Extensions to MARCdocs Database

Based on the above requirements, we propose to extend the database structure to store the following information:

- **Frequency count of all MARC content designation.** This data will be stored in a new database element that will contain an integer. Data for this element will be batch loaded from the frequency analysis spreadsheet reports
- **Indexable fields to support U.S. National Profile searches.** The data to be stored are the search labels in the U.S. National Profile. We know that the same subfield can support more than one search. For example, the 245\$a can be indexed to support the following searches:
 - Title Search – Keyword
 - Title Search – Keyword with Right Truncation
 - Title Search – Exact Match
 - Title Search – First Words in Field
 - Title Search – First Characters in Field

Therefore, for these searches, the following will be stored in a new database element:

- BP0.2
- BP1.5
- BP1.6
- BP1.7
- BP1.8

Initial data for this element will be batch loaded from the indexing guidelines document which identifies all candidate subfields that could be indexed for U.S. National Profile searches:

- BP0.1 (Author Search – Keyword)
- BP0.2 (Title Search – Keyword)

- BP0.3 (Subject Search – Keyword)
- BP0.4 (Any Search – Keyword)
- **Subfield tokens.** We propose to store tokens in the MARCdocs database. Tokens may consist of one or more strings of characters bounded by blank spaces. The form of the tokens that occur in the RadMARC records will be something like the following:
[MARCFieldTag][SubfieldCode]xxxxxx yyyy zzzzz

for example:

245axxxxxx yyyy zzzzz

However, there is no need to store the [MARCFieldTag][SubfieldCode] as part of the token in the database, since that can be generated based on the fact that each unique token will be associated with a specific subfield. Therefore, the form of a token stored in the MARCdocs database for a subfield that supports a specific search may look like the following:

xxxxxx yyyy zzzzz

Token Requirements to Support Profile Level 0 and 1 Searches for Author, Title, Subject

We discussed how a single token for a subfield could support the interoperability testing of more than one profile-defined search. Taking the example of the different title searches defined in the U.S. National Profile, we believe we can create a single token that can be used for testing each of the five title searches. In the following example, we will look at a token for the 245\$a.

1. To support testing of **BP0.2 Title Search – Keyword** we would need at least the following token in the database: **xxxxxx**. The token available for the Z-query could look like: **245axxxxxx**
2. To support testing of **BP1.5 Title Search – Keyword with Right Truncation**, the same token would be sufficient, as long as there was some logic in the test script creating the Z-query, for example, to truncate after the seventh character. The token in the database would be: **xxxxxx**. The token used for the Z-query could look like: **245axxx**
3. To support testing of **BP1.6 Title Search – Exact Match**, the same token could be considered sufficient, but not very useful. So, we would have a token with at least two character strings, so the following could be a token in the database: **xxxxxx yyyy**. The token used for the Z-query could look like: **245axxxxxx yyyy**
4. To support testing of **BP1.7 Title Search – First Words in Field**, the same token could be used as with **BP1.6**, but that wouldn't necessarily be useful. So, we would add another character string to the token and have: **xxxxxx yyyy zzzzz**. The token used for the Z-query could look like: **245axxxxxx yyyy**
5. To support testing of **BP1.8 Title Search – First Characters in Field**, we could use the token defined for BP1.7: **xxxxxx yyyy zzzzz**. The token available for the Z-query could look like: **245axxx**

Therefore, with some deliberation, we might be able to conclude that the token **xxxxxx yyyy zzzzz** could be stored with the 245\$a in the MARCdocs database, and this token will be sufficient to test all five profile-defined title searches. This may be too simplistic, but we propose this for discussion.

For the initial records, we may want to consider hand-crafting the tokens if the number of subfields that need to be populated is relatively small. But at some point, automatic generation of the tokens would be desirable.

Format of the Tokens

Discussions resulting from the first draft of this document yielded agreement among the project team on a structured and semantically rich token. The token elements will have the following component parts:

- A single alpha character for lefthand padding: Agreement to use the letter “r” for this character
- A single alpha character to indicate the format of the material being described or type of record: Agreement to use the code as defined in MARC 21 for Leader/06 – Type of Record as follows:
 - a - Language material
Includes printed, microform, and electronic language material.
 - c - Notated music
Includes microform and electronic notated music.
 - d - Manuscript notated music
Includes microform manuscript music.
 - e - Cartographic material
Includes maps, atlases, globes, digital maps, and other cartographic items.
 - f - Manuscript cartographic material
Includes microform manuscript maps.
 - g - Projected medium
Examples include: motion pictures, videorecordings (including digital video), filmstrips, slides, transparencies, or material specifically designed for projection.
 - i - Nonmusical sound recording
Examples include: speech.
 - j - Musical sound recording
Examples include: phonodiscs, compact discs, or cassette tapes.
 - k - Two-dimensional nonprojectable graphic
Examples include: activity cards, charts, collages, computer graphics, drawings, duplication masters, flash cards, paintings, photonegatives, photoprints, pictures, photo CDs, postcards, posters, prints, spirit masters, study prints, technical drawings, photomechanical reproductions, and reproductions of any of these.
 - m - Computer file
Includes the following classes of electronic resources: computer software (including programs, games, fonts), numeric data, computer-oriented multimedia, online systems or services. For these classes of materials, if there is a significant aspect that causes it to fall into another Leader/06 category, the code for that significant aspect is used instead of code m (e.g., vector data that is cartographic is not coded as numeric but cartographic). Other classes of electronic resources are coded for their most significant aspect (e.g., language material, graphic, cartographic material, sound, music, moving image). In case of doubt or if the most significant aspect cannot be determined, consider the item a computer file.
 - o - Kit
Contains a mixture of components from two or more types of items, none of which is the predominant constituent of the kit.
 - p - Mixed material
Indicates that there are significant materials in two or more forms that are usually related by virtue of their having been accumulated by or about a person or body. Includes archival fonds and manuscript collections of mixed forms of materials, such as text, photographs, and sound recordings.
 - r - Three-dimensional artifact or naturally occurring object
Includes man-made objects, such as models, dioramas, games, puzzles, simulations, sculptures and other three-dimensional art works and their reproductions, exhibits, machines, clothing, toys, and stitchery, and naturally occurring objects, such as microscope specimens and other specimens mounted for viewing.
 - t - Manuscript language material
- Three numbers indicating the Field Tag: As defined in MARC 21 specifications
- A single integer to indicate number of occurrence the Field Tag
- A single alpha character to indicate the Subfield Code: As defined in MARC 21 specifications
- A single integer indicating the offset within subfield: Using the following scheme: 1=first word in subfield, 2=second word in subfield; 3= third word in subfield, etc.
- A single integer indicating the version identification of the token: Differences in token versions may be based on Threshold of Occurrences, or some other discriminator

- A single alpha character for righthand padding: Agreement to use the letter “r” for this character.

And example token element to show this structure is: `rm2451a11r`. We can parse it as:

- r - lefthand padding
- m - type of record -- this is a Monograph-type record
- 2 - field code
- 4 -
- 5 -
- 1 - First occurrence of field in record
- a - Subfield code
- 1 - Offset within subfield -- 1=first word in subfield
- 1 - version identification or other discriminator
- r - Righthand padding

If there was a second instance of this field (in this example the 245), the token element for the second occurrence would be `rm2452a11r`. We can parse it as:

- r - lefthand padding
- m - type of record -- this is a Monograph-type record
- 2 - field code
- 4 -
- 5 -
- 2 - Second occurrence of this field in the record
- a - Subfield code
- 1 - Offset within subfield -- 1=first word in subfield
- 1 - version identification or other discriminator
- r - Righthand padding

The following is an example to show what sort of tokens (consisting of one or more token elements) might be created for an existing MARC record. The MARC record elements are listed in normal Courier font and the tokenized elements are in Courier bold:

```
LDR01019cam 2200265 4500^
Note: Leader/06 has a code of "a" which means the type of record is Language material
001ocm00000003^
Note: The 001 will contain a unique local record number and will not be tokenized.
0030CoLC^
Note: Note: The 003 will contain a code for the number in 001 and will not be tokenized.
00520010925133908.0^
Note: The 005 will contain the date/time stamp and will not be tokenized.
008690414s1963 nyu b 000 0 eng ^
Note: The 008 and other fixed fields will not be tokenized.
010 _a 63064323 ^
Note: The 010 contains the Library of Congress Control Number and will not be tokenized (and likely not contained in the RadMARC records).
040 _aDLC _cDLC ^
Note: The 040 contains a coded value for the cataloging source and will not be tokenized (and likely not contained in the RadMARC records).
05004_aHV700.5 _b.N37 ^
Note: The 050 contains a Library of Congress Call Number and will not be tokenized (and likely not contained in the RadMARC records).
```

0820 _a362.7/3 ^

Note: The 082 contains a Dewey Decimal Classification Number and will not be tokenized (and likely not contained in the RadMARC records).

1102 _aNational Study Service. ^

Token Elements: rm110a11r rm110a21r rm110a31r

24510_aIllegitimacy and adoption in Maine : _breport of a study made for the Maine Committee on Children and Youth. ^

Token Elements: rm245a11r rm245a21r rm245a31r rm245a41r rm245a51r : rm245b11r rm245b21r rm245b31r rm245b41r rm245b51r rm245b61r rm245b71r rm245b81r rm245b91r rm245b101r rm245b111r rm245b121r rm245b131r

Note: In this case the offset integer is more than one character since there were 13 words in the actual MARC 245\$a; in the RadMARC records, this value will also be 9 or less.

260 _a[New York], _c1963. ^

Token Elements: rm260a11r rm260a21r rm260c11r

300 _a24 p. ; _c28 cm. ^

Token Elements: rm300a11r rm300c11r

500 _aCover title. ^

Token Elements: rm500a11r rm500a21r

504 _aBibliographical footnotes. ^

Token Elements: rm504a11r rm504a21r

650 0_aIllegitimacy _zMaine. ^

Token Elements: rm6501a11r rm6501z11r

650 0_aAdoption _zMaine. ^

Token Elements: rm6502a11r rm6502z11r

7101 _aMaine. _bCommittee on Children and Youth. ^

Token Elements: rm710a11r rm710b11r rm710b21r rm710b31r rm710b41r rm710b51r

This is just an example. The RadMARC records' subfields will have a token comprised of no more than three token elements.

Using the Extended MARCdocs Data to Support RadMARC Record Creation

We propose to select data from the MARCdocs database to populate (either manually or in the future in an automatic way) the radioactive MARC records. Based on the discussion so far, the MARCdocs database will have appropriate data to assist in the creation of the RadMARC records.

The logic of this is as follows:

Query MARCdocs to find all content designation that supports a specific search, where the content designation has a frequency count that meets an identified threshold of occurrence, and produce output that lists the content designation and the token for each specific content designation.

An example query might be: Find all content designation that supports BP02. The output of this query would be reflected in the following table:

Occurrence	Field	Subfield
419641	245	A

Occurrence	Field	Subfield
169178	245	b
64433	440	a
62853	490	a
32535	505	t
30443	830	a
30378	505	a
30173	245	h
29833	740	a
29669	490	v
23558	830	v
20166	240	a
17556	246	a
15417	700	t
10031	240	l
7743	505	g
6662	222	a
5805	780	t
5674	730	a
5529	810	t
5523	210	a
4924	810	v
4471	785	t
4075	800	t
4031	130	a
2784	245	p
2587	830	p
2425	830	n
2405	700	n
2231	776	c
2063	440	p
2026	240	k
1924	800	v
1881	240	n
1846	700	m
1819	700	p
1644	240	m
1492	245	n
1481	710	t
1229	776	t
1137	240	f
1133	700	r
1099	130	l
1082	787	t
1011	246	f
1007	240	r
963	740	h
962	222	b
947	130	p
843	210	b
805	240	h
778	440	n
703	730	p

Occurrence	Field	Subfield
663	700	o
608	246	p
590	770	t
561	775	t
558	730	l
512	240	s
496	730	f
463	130	f
429	240	o

Now, if we refine the query to ask only for the content designation that meets a specific threshold of occurrence, we could state it as: Find all content designation that supports BP02, where the occurrence of the content designation is greater than 100,000. The result would be:

Occurrence	Field	Subfield
419641	245	A
169178	245	b

We could decide to increase the threshold to greater than 50,000 and the result would be the following set of content designation:

Occurrence	Field	Subfield
419641	245	A
169178	245	b
64433	440	a
62853	490	a

This approach is attractive since we could rationally base our RadMARC contents to reflect empirical data from our previous content designation utilization. And it would allow us to work with a smaller number of fields/subfields.

In this case, we would create a MARC record that has only these four fields/subfields to populate. It would be relatively easy to handcraft the tokens for these and input them in the MARCdocs. Let's assume the following tokens associated with each of the fields/subfields (we don't know if these are the appropriate character strings for the tokens, but we will use them for this example):

- 245 \$a: xxxxx yyyy zzzz
- 245 \$b: qqqq rrrrrr sssss
- 440 \$a: aaaa bbbb ccccc
- 490 \$a: dddd eeee ffff

We could now do the query of the MARCdocs database: Find all content designation that supports BP02, where the occurrence of the content designation is greater than 50,000. And the output would be structured as follows:

Occurrence	Field	Subfield	Token in MARCdocs
419641	245	a	xxxxx yyyy zzzz
169178	245	b	qqqq rrrrrr sssss
64433	440	a	aaaa bbbb ccccc
62853	490	a	dddd eeee ffff

With such a structured output, it would be possible to transform the token into what would go into the appropriate RadMARC record field:

245axxxx yyzz
 245bqqqq rrrrr sssss
 440aaaaa bbbb ccccc
 490adddd eeee ffff

In addition to the content designation selected from the MARCdocs that could support the specific profile-defined searches, it will be necessary to populate other MARC fields that are likely to be expected by a system such as the 001, 005, 010, etc. We will also populate a specific field in the RadMARC record that would indicate the version of this record (e.g., putting the following in a note field: RadMARC record for threshold of occurrence greater than 50,000) which allows us to control the versions of the RadMARC records. In addition, we will need to set the indicator values for the fields contained in the specific RadMARC records.

Using the Extended MARCdocs Data to Support Z-Query Creation

The same data could be used for the test script that will create the various Z-queries necessary to test interoperability. For example, we could provide to the test script the following:

Profile Search	Field	Subfield	Token in MARCdocs
BP02	245	a	xxxx yyzz
BP02	245	b	qqqq rrrrr sssss
BP02	440	a	aaaa bbbb ccccc
BP02	490	a	dddd eeee ffff

The test script could read the data structured in this table, recognize that this is to create test searches for the BP02 searches, and then create the appropriate four Z-queries:

(1,4)(2,3)(3,3)(4,2)(5,100)(6,1) 245axxxx yyzz
 (1,4)(2,3)(3,3)(4,2)(5,100)(6,1) 245bqqqq rrrrr sssss
 (1,4)(2,3)(3,3)(4,2)(5,100)(6,1) 440aaaaa bbbb ccccc
 (1,4)(2,3)(3,3)(4,2)(5,100)(6,1) 490adddd eeee ffff

The output of the MARCdocs database query could easily help automate the creation of the Z-queries that would match the tokens in the query that are in the RadMARC record.

Summary and Conclusion

We think that the above approach to extending the MARCdocs database seems promising to help automate the RadMARC record creation and the Z-query creation. We want to make sure we have the tokens in the records that will be the search terms in the Z-query. The approach described seems to offer that.

The main thing will be to determine what the token recorded in the MARCdocs database for each field/subfield looks like. Thoughts on that are welcome.