2. BACKGROUND INFORMATION

2.1 MARC21 FORMATS

In the United States, there are MARC21 formats for bibliographic, authority, holdings, classification, and community information data. This document covers indexing of bibliographic data only. It includes no standards for indexing data from other MARC21 format records, or from other types of records in on-line systems for which no MARC21 format exists, e.g. order records. Nor does it include standards for indexing data officially defined as part of the MARC21 holdings format that is often included in bibliographic records, e.g., locations and call numbers.

The standards do include some obsolete MARC fields and subfields because they are still present in older records, as well as some fields and subfields that are used only in records cataloged using older (pre-AACR2) cataloging standards.

2.2 OTHER RELEVANT STANDARDS

Information Retrieval Service Definition and Protocol Specifications for Library Applications (ANSI/NISO Z39.50) defines protocols for intersystem searching of records. When an automated library system has a built-in Z39.50 client, a searcher can use the features of the “home” system to query another Z39.50 server and retrieve and display results as if the search were done in the home system. The searcher need not learn multiple protocols for searching multiple systems.

Even when ANSI/NISO Z39.50 has been implemented in multiple systems, the results of searches across those systems may differ widely if different data elements are included in the systems’ indexes. The intent of “Indexing Standards and Guidelines for Bibliographic Records” is to promote greater standardization of content of indexes in regional systems to provide searchers with more predictable results.

2.3 HOW SEARCHING WORKS

Library database searching involves a number of components and processes. A basic understanding of these components and processes can make the reasons for indexing decisions clearer.

The data in MARC records are labeled with field and subfield codes that enable the machine to identify different data elements for indexing. Indexing programs select designated data elements and add them to designated indexes. Heading or phrase indexes are designed to contain strings of elements in an established order, usually the order presented on the catalog record. Keyword or word indexes are designed to contain individual words, sometimes embedded in a phrase to permit proximity searching for two terms together.
There are also two types of searches that can be directed at indexes: scan or browse searches, and find or “search” searches.

Scan or browse searches are intended to return lists of access points (e.g., names, titles, subject headings, or classification numbers) to the user. The rules that determine how these data elements are entered in the catalog record also work to create an ordered list when the elements are searched using a scan or browse search command. The retrieval of brief or full bibliographic records is an additional step when doing a scan search. The initial result is an ordered list of access points and not simply a set of records. In some systems, “index records” are constructed containing the specified indexed string and additional data from the bibliographic record specifically for searching, sorting, and display. The browseable list of access points or index records may extend beyond the terms used in the original search, allowing the user to scan all the headings in the index one screen at a time.

Find searches are intended to return a set of records that share a common data element or elements. The searched terms can be specified as coming from particular parts of the record, depending upon how the indexes have been defined. The terms can also be combined using Boolean operators to construct sets with more complex logical definitions. The primary result of a find search transaction is a set of records. It does not produce a browseable list of headings, although one may be able to sort a displayed list of brief records by certain key elements (e.g., title proper, main entry author, publication date). Because find searches are typically done using keyword indexes, they are sometimes referred to as keyword searches.

Depending on how one’s system has been programmed and configured, it may be possible to combine either type of search with either type of index.

A scan or browse search can be performed on a heading or phrase index to produce a list of headings; but it can also be performed on a keyword index to produce a list of the keyword terms contained in the index.

A find search can be performed on a keyword index to produce a set of records containing the sought term; but it can also be performed on a heading or phrase index to produce a set of records containing the specified heading or phrase. The set can then be combined using Boolean operators with other sets generated by find searches.

Library systems typically ask the user to specify a type of search (browse or find), a type of data (author, title, subject), and a search value (the word or phrase being sought). Sometimes the names of indexes combine a search type and a data type, e.g., “Author begins” vs. “Author keyword.” In find searches, Boolean operator values are also specified. The system reconfigures the user’s search into a system command that specifies the type of search, index name, and search value using a syntax that the system’s programming understands. A similar reconfiguring of the information the user enters is done with search qualifiers. The user selects qualifying terms of a given type (e.g., date, format, language) from a system-provided menu (e.g., for language, English, French, German). The system then internally translates the user’s selection into the appropriate index names and encoded values (e.g., translating the language qualifier “French” into a search of the appropriate keyword indexes of 008 and 041 field positions for the value “fre”) and combines this search with the rest of the search using an appropriate Boolean operator.
The extent to which two systems can produce similar results when given a particular search will depend upon how comparable the indexes defined for each system are, and on how successful the configuring of Z39.50 clients and servers to translate between systems has been. For these reasons, this standard makes recommendations regarding which indexes each system should include to support networked regional searching using the Z39.50 protocol.

2.4 RECALL, PRECISION, AND MEASURES OF SEARCHING SUCCESS

Both recall (retrieval of all potentially relevant records) and precision (retrieval of relevant records only) are important to searchers. One may be more important than the other in any given search. A searcher who wants comprehensive results will appreciate high recall and may be willing to sort through some extraneous records in order to meet that goal. A searcher who wants to retrieve a single known record will appreciate high precision. Different types of searches are designed to meet these varying needs.

Traditional measures of precision and recall apply primarily to find search results. They do not account for additional factors such as ranking. Good ranking can reduce the negative impact of poor precision in a large result set. Browse searching success is measured by comparison to an ideal filing order and by the context provided for the heading sought. Measures of searching success should take into account the nature of the indexing as well as the searcher’s goal, skill, and level of satisfaction.

2.5 INDEXES

2.5.1 BUILDING INDEXES

An index to bibliographic records is a set of data extracted from specific fields, subfields, and bytes within the bibliographic record. In most online systems, search strings are actually matched against index entries, rather than against the bibliographic records themselves. Links from index entries to the bibliographic records in which the indexed data appears allow the searcher to move from the index entries to the records themselves.

2.5.1.1 Normalization of Index Entries

Because of the variety of ways in which information in a bibliographic record might appear (e.g., with or without capitalization, with variations in punctuation), index entries should undergo the process known as normalization to create a standardized form for the entry. Common steps in normalization include the conversion of all alphabetical characters to upper (or lower) case; removal of punctuation marks, extraneous spaces, and diacritical marks; and conversion of special characters to a standard alphabetical equivalent (e.g., the digraph “æ” would be converted to the separate letters “ae”). For one example of normalization rules, those followed by the Library of Congress in creating its online LC/NACO Authority File (LNAF), see <http://lcweb.loc.gov/catdir/pcc/naco/normrule.html>.

Normalization routines do vary significantly from system to system, both for index entries and for search queries. Ideally, a system will normalize a search query for the user before
comparing it to index entries, rather than requiring the user to perform the normalization by omitting punctuation or some other data. For example, a searcher may enter the title “stocks & shares,” but the index entry for this title may be “stocks shares” because, in the particular system, the ampersand has been removed during normalization. That system then should also remove the ampersand in the search string before processing the search, rather than returning incorrect results or no results because the ampersand is present.

2.5.1.2 Purposes for Indexing Specific Data Elements

There are two major purposes for which data elements are indexed:

- retrieval: these are the data elements that a searcher is likely to include in a query;
- sorting: these are data elements that a searcher is unlikely to include in a query but that are needed for intelligible and useful sorting of index entries for display.

For example, a searcher who wants to find titles by a particular personal author will usually include the person’s surname and forename in the search string. If s/he is searching for works by a particular ruler, s/he may well include numeration, e.g. “George III,” rather than simply “George.” S/he probably will not include the person’s birth and death dates, and is very unlikely to include qualifying information; e.g. s/he is much more likely to search for “Eliot, T. S.” than “Eliot, T. S. (Thomas Stearns).

However, if some of the data elements that are unlikely to be included in search strings are not included in index entries, the searcher may have to sort through many extraneous entries to find the relevant ones.

Contrast the following hypothetical search results. In the first example, personal authors’ birth and death dates have not been included in the index:

<table>
<thead>
<tr>
<th>Number of Titles</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>Jones, John</td>
</tr>
</tbody>
</table>

In the second example, dates have been included:

<table>
<thead>
<tr>
<th>Number of Titles</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Jones, John</td>
</tr>
<tr>
<td>5</td>
<td>Jones, John, 1879-1939.</td>
</tr>
<tr>
<td>10</td>
<td>Jones, John, 1920-</td>
</tr>
</tbody>
</table>

2.5.1.3 Stopwords

To reduce the size of indexes and the time required to process searches, index designers often create a list of stopwords, i.e., words that simply are not indexed.
Typically, words used as operators by the system have been stopwords in keyword searches because the system could not determine whether these words in search arguments should be processed as operators or search words. Although this is not a problem with a well-designed graphical interface, in which the searcher enters operators and search terms in separate parts of a dialog box, most systems still have both graphical and command driven interfaces. It is impractical to create multiple indexes for multiple interfaces, so the limitations forced by the command driven interface still apply.

Lengthy stopword lists are very difficult for searchers to remember. They also make searches on some topics difficult or impossible. The standards defined in this document therefore require that there be no stopwords in scan or browse searches other than initial articles in title phrase searches and only a minimum number of stopwords in find searches.

Unfortunately, some systems return incomplete or confusing results, or no results at all, to queries that include stopwords. The standards defined in this document therefore state that stopwords other than operators in keyword search arguments should be ignored, i.e., the search should be processed as if the stopword(s) were not present, and a message to that effect should be returned to the searcher.

2.5.2 TYPES OF INDEXES

2.5.2.1 Phrase Indexes

Phrase indexes (also known as exact or heading indexes) include all the normalized data within the field or subfield indexed. To browse entries in these indexes, the searcher must enter a browse search string that begins with the first indexed character in the first indexed subfield and includes all indexed characters in the same order in which they appear in the bibliographic record (note the explanation of normalization of index entries in section 4.a.1 above). When the searcher has sufficient information to browse search a phrase index, both precision and recall can be excellent.

2.5.2.2 Keyword Indexes

Keyword indexes include all words other than stopwords from indexed fields and subfields. A “word” is typically defined as any string of alphanumeric data preceded by and followed by a space following normalization. To access records using keyword indexes, the user may enter any indexed word in any order. Find searches of keyword indexes generally increase recall, especially when they combine synonyms or use truncation. Precision may be low unless distinctive terms or qualifiers such as date or language are used.

2.5.2.3 Number Indexes

Number indexes include standard numbers assigned by national agencies, such as ISSN or ISBN; system-assigned numbers such as record identifier; and locally assigned numbers of various sorts. Generally searchers who use number indexes expect a very precise result, usually retrieval of a single record.
2.5.2.4 Qualifiers

Qualifiers are data elements used to limit find search results to records meeting specified criteria, such as date of publication, language, or physical format. Typically, the data elements indexed are coded data from fixed-length fields. When used in conjunction with Boolean operators, qualifiers can significantly improve precision of find search results.

2.5.3 Accessing Data in Indexes

The result of a scan or browse search in a phrase index is typically a list of index entries of a particular type with a highlighted index entry that is closest to the character string the searcher entered. Theoretically, the searcher can either browse through all the index entries matching the search if only a segment of the full index has been retrieved, or move backward in the index to its beginning or forward to its end. Browsing index entries often aids the searcher in determining which search terms to use or how to refine a search. In library systems, author, title, and subject phrase indexes are most likely to be browseable; keyword indexes are sometimes browseable as well.

2.5.3.1 Truncation

Truncation allows the searcher to specify a partial match between search terms and index entries in order to increase recall. Truncation may be automatic, in which case it may be possible to override, or it may be invoked through the use of a truncation symbol. Some systems allow the searcher to specify the number of characters represented by the truncation symbol. Left, right, and internal truncation are possible. Internal truncation is also referred to as use of “wild cards” or “masking.”

Since processing searches that include truncated terms generally places a significant load on the system, designers often place limits on the use of truncation or types of truncation. More than one type of truncation of a single term frequently is not possible. Some systems even limit the searcher to a single instance of truncation in a search argument.

2.5.3.2 Operators

2.5.3.2.1 Boolean Operators

Boolean operators specify logical relationships between indexed terms and typically appear between two find searches, indicating how the results of the two searches should be combined. They include:

- **AND** (the terms from both searches must be present in the retrieved records);
- **NOT** (the first term must be present in the retrieved records, but the second term cannot be);
- **OR** (either term or both terms must be present in the retrieved records);
- **XOR** (either term, but not both terms, must be present in the retrieved records (also called Exclusive Or)).

Use of AND, NOT, and XOR improve precision; use of OR improves recall.
2.5.3.2.2 Positional Operators

Positional operators specify the proximity of two terms in index entries for a record. They include:

- **ADJ** The terms must be adjacent to each other in the specified field or subfield indexed, i.e., the second must immediately follow the first.
- **SAME** The terms must be in the same specified indexed field or subfield, but may be in any order and need not be adjacent. Note that some systems consider all fields with the same tag to be the same field.
- **WITH** The terms must be in the same sentence in the specified indexed field or subfield. In most systems, a sentence is defined as the text from the beginning of a field to a period or between two periods.
- **NEAR** The terms must be adjacent to each other in the specified field or subfield indexed, but may be in either order, or the terms must be within a user-specified distance (usually number of words) of each other in the specified field or subfield indexed, in either order.

2.5.3.2.3 Default Operator

One operator is usually chosen as a default operator for find searches; i.e. if the searcher does not specify an operator in a find search that queries multiple terms in a single index, or a find search that combines queries of multiple indexes, the system processes the search as if the default operator were present.

- **AND** is the most common default operator.

2.5.3.2.4 Multiple Operators and Nested Searches

In systems that allow the use of multiple operators in a single search, operators are typically processed in a predetermined order, regardless of the order in which they are entered, unless the searcher overrides the default order. Usually, this is done by placing parentheses or some other symbol around the search terms connected by a particular operator. The system then processes find search of the strings within parentheses before the remainder of the search argument. In especially complicated cases, parentheses within parentheses may be used; these are called “nested searches”. 