**Cleaning up Bible headings**

Gary L. Strawn

23 Nov. 2010

**Introduction**

Conventions for constructing headings for the various parts of the Bible, and the versions or translations in which they appear, have changed with time; most large bibliographic databases contain Bible headings that reflect a variety of practices. The advent of MARC coding did nothing to simplify the matter: some institutions did not apply any internal subfield codes to Bible headings (especially during retrospective conversion), and some applied incorrect subfield codes. To be sure, many institutions rigorously applied the correct subfield codes; but the point remains that a large database will again reflect a variety of practices.

The lack of subfield coding, or (in most but not all cases) the use of incorrect subfield codes, has not by this fact alone necessarily impeded user access—at least not when access was sought via keyword searches or left-anchored heading searches—because the proper words were often present, and often in the proper order. But it appears that the world is about to change. One of the large-scale alternations that will need to be made to existing headings in bibliographic databases at the time of any adoption of RDA is a change to the structure of a large number of headings for parts of the Bible. (Specifically, the designations for *testaments* will disappear from the headings for books or groups of books.) Such a change will be effectively impossible if the headings in a database are not already in current form, and especially if they are not tagged with proper subfield codes. If some effort is made now to bring existing headings into their current form (including proper MARC content designation), not only will access be improved (for some headings) for the present, but any transition to the RDA forms of headings will be vastly simplified.

At Northwestern University Library, the effort to evaluate and modify existing Bible headings has of late been the subject of intense work. Most of this work has involved the development of a "Bible cleanup" program to examine the Bible headings in a Voyager database (130, 63, 730, 830 fields) and sort them into three categories:

1. headings that appear to be correct as they stand
2. headings that are not correct but can automatically be modified to the putative correct form
3. headings that appear not to be correct and must be made subject to operator review

This program has developed to the point where it can be tried out at other Voyager institutions interested in the cleanup of Bible headings. The program needs to be tested at institutions other than Northwestern to establish a reasonable level of confidence that it performs its tasks correctly and reliably. In addition, other bibliographic databases may contain categories of problem Bible headings not present in Northwestern's database, and these new problems may call for enhancements to the program so that the program is as useful as possible to as wide an audience as possible.

It should be noted from the outset that although the program for examining Bible headings has received a lot of attention, it should not yet be used as a production program to change records in a Voyager database. The program has been run successfully as a production job against Northwestern's test database (which is a copy of its production database), but the program has not yet been used as a production program in the production database at Northwestern, and should not be so used at other institutions until it has seen additional testing.

**Installation and general configuration**

Create or identify a folder on some handy drive to contain the reports generated by the program; this folder can be shared with other programs. Also create or identify a folder on some handy drive to contain the program's configuration files; this folder can be shared with other programs, but should not be the same as the folder for reports.

The installation package for the Bible cleanup program can be found here:

 <http://files.library.northwestern.edu/public/Bible>

During the testing phase of program development, you won't be using the program for production work; so you don't need to make sure that the BatchCat version number of the installation package matches your local BatchCat version number. Once the program is approved for production use at other institutions, Northwestern will supply appropriate installation packages for relevant BatchCat versions. Download the ZIP file for BatchCat version 2007.22.416 (Northwestern's current version), unzip it, and run setup.exe in the usual way.

The download site also contains a set of configuration files (with the extension .cfg), which you should also download. Copy these configuration files into the folder you created to contain them.

If you do not at present use the cataloger's toolkit program, go to this location:

 <http://files.library.northwestern.edu/public/CatalogersToolkit/>

Open the file CtkSupport.ZIP and copy the file called codes.cfg into the folder you identified for configuration files.

Start the program and go to the options panel. Fill out the options panel in the usual way. The panel contains one options work area that is not common to many of the other programs prepared at Northwestern University Library. The *Folder for Bible configuration files* box contains the name of the folder you defined to hold the program's own configuration files.

**Configuration files**

The program draws on a series of configuration files to do its work. Default versions of all of these files are available at the Northwestern University Library download site; it is strongly urged that you use these files as the basis for the work at your local institution, rather than subject yourself to the tedious labor that would be required to construct them from scratch. You will modify these base files as necessary to match conditions in your local database. These configuration files are plain text files, and can be created and modified with the Windows Notepad or any other suitable program.

During testing, you should probably start with the files as given. As you review the lists of problem headings created by the program, and also the list of changed headings, you will probably find that you need to make some changes to these files (most of the time these will be additions). After you change the configuration files you should perform another test to verify the effect of your changes.

Each configuration file contains a header describing the purpose of the file, and the layout of its contents. In all of these files, use the pound sign (#) as the first character in a line to identify comments.

*BibleCleanupAmbiguousBookLanguageVersion.cfg* Most books of the Bible, languages, and names of particular versions have distinct names, but a few names are shared in more than one category. For example, *Daniel* is the name of the book of the Bible, and *Daniel* is also the name of an editor; *Moore* is the name of an editor, and *Mooré* is the name of a language. This configuration file lists words that fall into more than one category (book, language, version), and identifies the subfield appropriate to each such word. For example, to define *Marshall* as the name of a language and the name of a version, include these two lines in this file:

 $l Marshall

 $s Marshall

*BibleCleanupBiblsToSkip.cfg* This file lists any bibliographic records that the program should not subject to any attention. This is the only configuration file that is not required; if not present, the program will not attempt to generate it. This file contains a simple list of such Voyager bib record numbers. As you test the program, you will probably identify a few records that are formulated correctly but which are reported as problems. If this matter cannot be resolved, you can list those records here; in future runs of the program, the program will skip these records altogether.

*BibleCleanupBook.cfg* This file lists all of the recognized books of the Bible, and significant variations on them; it also lists the names of groups of books. The names of the two testaments themselves (with variants), and the Apocrypha, are hard-coded into the program; don't attempt to list them here. Each line in the file contains the name for the book as it might be found in bibliographic records, the correct name for the book (only supply this element if the first name is not the correct name), plus a series of true/false flags:

* book of the Old Testament
* book of the New Testament
* book of the Old Testament Apocrypha
* is citation of chapter and verse allowed?
* complete replacement flag (no longer used; always False)

For example, this line (with tabs shown explicitly for clarity) defines *Kings, 1st* as the accepted name for a book of the Old Testament:

 Kings, 1st<tab><tab>True<tab>False<tab>False<tab>True<tab>False

This line (again, with tabs shown explicitly for clarity) defines *I Kings* as a variant name for a book of the Old Testament; when found this should be changed to *Kings, 1st*:

 I Kings<tab>Kings, 1st<tab>True<tab>False<tab>False<tab>True<tab>False

Give definitions for collections of books of any part of the Bible after the definition of individual books found in that part. For example, define collections of books in the Old Testament after the definitions of all individual books in the Old Testament.

*BibleCleanupLanguage.cfg* This filelists the names of exceptional languages that may appear in subfield $l of Bible headings. It is only necessary to list in this file those languages that are not also listed in the cataloger's toolkit's codes.cfg file—basically, this means that you only need to list in this file those language names that cannot be equated directly to an individual 3-character MARC language code. (For example, you will need to list a language name in this file when the language is represented only by a MARC code for a group of languages, and when the name of an earlier form of a language contains a parenthetical addition. Each line in the file consists of two elements: the name of a language as it may be found in bibliographic records, and (when this is not the correct name) the name to be used instead.

For example, this line (with tabs shown explicitly for clarity) defines *Dutch (Middle Dutch)* as a valid language name:

 Dutch (Middle Dutch)<tab>

This line (again, with tabs shown explicitly for clarity) defines *Anglo-Saxon* as a variant name that should be changed to *English (Old English):*

 Anglo-Saxon<tab>English (Old English)

*BibleCleanupLcshAuthors.cfg* This file lists the names of groups of authors that may be used in subfield $x in certain LCSH headings. (For example, following the subdivision *Meditations*.) Each line in the file lists one such subdivision in its entirety.

*BibleCleanupLcshDenomination.cfg* This file lists the names of denominations that may be used in subfield $x in certain LCSH headings. (For example, following the subdivision *Study and teaching.*) Each line in the file lists one such denomination name in its entirety.

*BibleCleanupLcshLanguageGroup.cfg* This file lists the names of language groups that may be used as part of certain phrase headings. (For example, in subfield $x following *Versions.*) Each line in the file lists one such name of a language group.

*BibleCleanupLcshNotProphecies.cfg* The LCSH subdivision *Prophecies* may be followed by any valid subject heading, to describe the content of an item that deals with Biblical prophecies on that subject. This file lists subfields that may appear after the subdivision *Prophecies* that should *not* be considered the subject of Biblical prophecy, but should be evaluated according to their own merits. (For example, a line in this file should list the subdivision *Bibliography* so that the program construes a heading with the construction *… $x Prophecies $v Bibliography* as describing a bibliography of prophecies and not as as containing prophecies on the subject of bibliography.)

*BibleCleanupLcshSubdivisions.cfg* This is the main file that controls the program's work with LCSH form, topical, chronological and geographic subdivisions. Each line in the file identifies a single subdivision, or variant name for a single subdivision, or a string of valid subdivisions. A line in this file that describes a valid subdivision or string of subdivisions contains the appropriate subfield codes and subfield texts; a line that describes the variant name for a valid subdivision has the same form, but begins with the reserved text REFER FROM. Any line that describes the variant name for a valid subdivision must immediately follow the line that describes the valid form for that subdivision; any valid subdivision may have multiple variant forms. If a valid heading may be followed by a geographic subdivision, add *(May Subd Geog)* to the end of the subdivision's text.

For example, this line defines *Catalogs* as a valid form subdivision:

 $v Catalogs

This line defines *Collectors and collecting* as a valid topical subdivision; this subdivision may be followed by one or more geographic subdivisions:

 $x Collectors and collecting (May Subd Geog)

These lines define *Commentaries* and the combination *Commentaries—History and criticism* as valid topical subdivisions, and *Commentaires* as a variant form for *Commentaries:*

 $v Commentaries

 REFER FROM $v Commentaires

 $x Commentaries $x History and criticism

*BibleCleanupVersion.cfg* This file lists named versions of the Bible and parts of the Bible, as they should appear in subfield $s of Bible headings. Each line consists of the name as it may be found in subfield $s, the proper formulation (if the name as found is not the correct form), the name of the language with which this version is associated, and a true/false indicator that identifies the names of repositories.

For example, the following line (with tabs shown explicitly for clarity) defines *Ailmer* as the name of a Greek version of the Bible:

 Aaronson<tab>English<tab>False

The following line (again, with tabs shown explicitly for clarity) defines *Fisher and More* as the variant form for *Fisher-More:*

 Fisher and More<tab>Fisher-More<tab>Latin<tab>False

Although each version is associated with a language in this file, the program does not at present make use of this information to guarantee that a version name only appears with an appropriate language.

The following line identifies *British Library* as the name of a repository. During validation of a Bible heading that contains this text, the program will expect subfield $s to be followed by subfield $k containing *Manuscript* which is in its turn followed by subfield $n containing the designation for a manuscript at the British Library. Note that no language is associated with the names of repositories.

 British Library<tab><tab><tab>True

**Use of the program**

When you are ready to test the program, start the program and select Perform from the program's menu. The program sets about its work, and requires no intervention of any kind from you. When the program has finished its work, it will cancel itself.

Here's an outline of what happens:

1. The program asks Voyager for a list of all authority-controlled headings that begin *Bible*. (The program asks for every line in the Voyager HEADING table that begins with those five letters.) The program makes a list of these headings, creating a snapshot of the state of Voyager headings at a moment in time. (This temporary file is called Scratch.$$$; you can look at this file if you want. The program re-creates this file every time you run it.)
2. The program then reads through this list of headings, and for each retrieves each bibliographic record that contains the heading. (The program only examines a given bibliographic record once during a given run.)
3. The program submits each 130, 630 (with second indicator "0"), 730 and 830 field that begins *Bible* and performs a series of tests and operations on it. As a result of this work, the program will have changed the heading in the bibliographic record, declared the heading to be acceptable as found, or reported it for review.
4. After reviewing all of the headings in one bibliographic record, during a *test* run the program writes the changed record to a file for your examination. During a *production* run the program writes the changed record back to Voyager.

As the program does its work, it shows you a panel with running tallies of headings and bibliographic records examined and changed. This busy display is mostly so you can be sure that the program has no become stuck. (If the program *does* appear to have become stuck, take a picture of this panel (the display contains the current bibliographic record number, which will be an important clue for diagnosing the problem) and then cancel the program.)



**Reports**

The program produces a set of report files. The names of these files begin "BIBLECLEANUP" and contain the date and time you selected Perform from the program's menu. The following description of each file shows the part of each the file's name after the date and time stamp.

*BibleNotBible.txt* This file lists headings that the program examined but which it determined during the course of its work *not* actually to be a Bible heading. This is largely just for your information.

*Fyi.txt* This file lists changes made to bibliographic records that are just for your information. Most of the changes listed will involved the addition of subfield $f to 130 and 730 fields for specific versions that lack subfield $f.

*HeadingsChanged.txt* This file lists each bibliographic heading that the program changed—even if the only change is an adjustment of punctuation. Each line in the tab-delimited file contains:

* Voyager bibliographic record number
* Former tag
* Former indicators
* Former field text
* New tag
* New indicators
* New field text

One way to review this file is to import it into Microsoft Excel or Microsoft Access and sort it by either the former or new field text, and systematically compare the old and new forms of each heading.

*OkAsIs.txt* This file lists each heading that the program believes to be already in the proper form. Review this to see if you agree with the program's judgment.

*PairedHeadings.txt* This file lists headings that the program believes are redundant, and identifies the heading the program will remove during a production run. Headings listed in this file are pairs of headings with compatible tags that contain the same words, but in a different order. (The program does not include headings that contain the text "relation to" when drawing up this list.) One important source for such headings lies in the practice in the past of creating a second access point for a topic or version to provide a second way to find bibliographic records. (These headings were discontinued long before the arrival of MARC coding, which means that there is no agreed-upon way to code them.) Here are a couple examples of headings that will be reported in this file:

 630 00 $a Bible. $p N.T. $x History of contemporary events.

 630 00 $a Bible. $x History of contemporary events $x N.T.

 130 0 $a Bible. $p O.T. $p Psalms. $l English. $f 1932.

 730 0 $a Bible. $l English. $p O.T. $p Psalms. $f 1932.

During retrospective conversion, or perhaps in some cleanup following conversion, some of these headings were flipped around to the correct form without a check for duplication in another heading. This means that you will also find reports of headings that are absolute duplicates of each other:

 630 00 $a Bible. $p Gospels $x Criticism, interpretation, etc.

 630 00 $a Bible. $p Gospels $x Criticism, interpretation, etc.

During testing, if you agree with the program's decision on which heading should be deleted, you need do nothing. If you disagree, the best thing is for you to adjust the bibliographic record yourself. The program will be able in most cases to decide which heading is redundant and should be deleted; if the program cannot decide (usually this is because it believes that both of the headings as given contain errors) you must make adjustments yourself.

*Problems.txt* This file lists each heading that contains a problem identified by the program. When the program identifies one problem with a heading it stops work on the heading; a heading may contain problems beyond those identified by the program. Each line in this file contains a brief message, the text of the original heading, and the Voyager bibliographic record number. You will probably find it useful to import this file into Microsoft Excel, Microsoft Access, or some other program and sort it in alphabetical order—you will often find it easier to deal with all occurrences of a particular problem at one time.

*RecordsModified.txt* This file lists the final version of each bibliographic record that the program will update during a production run. (The program only generates this file during a test run.) If you review the list of changed headings and the *FYI* file, then you probably don't need to review this file as well.

*Report.txt* This file contains a simple statistical summary of the program's work.

*SubfieldOrderNotEvaluated.txt* This file lists problems of a different type than those listed in Problems.txt. After the program has parsed a heading and believes it has found something to change, it attempts to put the subfields of the changed heading into what it believes to be the proper order. Some problems surface at this point instead of during the parsing. For example, the parser will allow a heading to have, or to acquire, two $l subfields, or four $p subfields; but the routine that attempts to put subfields into order will recognize conditions such as these as a problem, and will report them for your attention. It is important to note that the heading presented in this file is the heading *as modified by the program,* and not the original heading in the bibliographic record. Careful attention to subfield codes assigned to the parts of the heading will often allow you to identify the source of the problem. Some of the problems reported in this file are the result of the programs failure to deal properly with ambiguous conditions. Regardless of the source of the problem, though, perhaps the best method for dealing with headings reported in this file is to make whatever adjustments are needed to the heading in the Voyager bibliographic record—and to any other Bible headings in the same record (remembering that the original heading may well already be correct), and then to add the record's Voyager bibliographic record number to the *BibleCleanupBibsToSkip.cfg* file, so the program doesn't attempt to change the record on a subsequent run.