

Using the XC OAI Toolkit Librarian Conversion Log for MARC records

If a MARC record fails during the conversion process, the record number and an error message will be written to the log ("librarian_convert.log"). The actual error messages in the log are supplied by the MARC4J software used within the Toolkit, and may or may not be helpful in figuring out what the actual error may be in the record! We have found that the OAI Toolkit will sometimes tolerate a small number of errors in a file being loaded, but beyond that the errors in some records may cause other records that come after the problem record to fail even when those records do not have obvious problems. Finding and correcting errors in just a few records within a file may make the entire file process correctly, and is much less overwhelming than attempting to correct every error individually.

Convert Process: Suggested Debugging Methods

You may want to start by creating a separate file of the record that produced the first error in the log - and the record BEFORE it - and test just those two records. This may catch situations where one record with a problem has a cascading effect on the records that come after it. If ALL of the records after a certain point fail, this is a good technique to try.

The record number for the first record that failed can be found in the log file. If you are doing a full extract of all of the records from your ILS "in order", it is likely that the previous record in the file has the record number one less than that of the record in the log - if not, you may have to look back at the file being processed to obtain the number for the previous record. One way to view the raw MARC file is by using MARCEdit (<http://oregonstate.edu/~reaset/marcedit/html/index.php>), a free metadata viewing/editing tool that has a variety of record debugging uses. After running the input file through the MARCBreaker feature of MARCEdit, search for the record number of the first error in the log, and then backtrack through the file to the previous record to get its record number.

If this does not solve the problems, or the errors don't follow this pattern, you may want to create a separate file of just the records that fail the conversion process by using the record numbers from the log to identify them. A sample script file to extract the error record numbers from the "librarian_convert.log", called "extract_error_recordno.bat/sh", is written in Windows/Linux and is included in the respective Windows/Linux installers. Correcting the errors in the records may be an iterative process, and in this way you can work with smaller and smaller files of just the records that continue to fail the conversion process. Look for patterns or similar errors that happen frequently. Once the most prevalent errors are corrected it will be easier to tackle the others in a sequential manner.

When you find errors in MARC records, the best solution is to correct the errors in the original repository (e.g. the ILS) if that's possible, so the error won't be a problem if the record is harvested again. However, this may not be feasible during the debugging process. One interim strategy is to perform some cleanup of the data in-between export and conversion/load with a perl script using

MARC::Record. It may also be helpful to edit the records that fail by using an external tool such as MARCEdit. If you do correct the records using MARCEdit or another external tool during the debugging process, you will either need to reload the corrected records back into the original repository so that they can be reprocessed by the OAI Toolkit, or correct them again in the original repository. (In other words, don't load them from the external tool directly into the OAI Toolkit, or the Toolkit won't be able to track them properly).

As a troubleshooting mechanism, you can also use MARCEdit or another tool to convert the MARC records that fail to MARCXML, and then view the MARCXML for possible parsing errors. However, these MARCXML records should NOT be loaded into the OAI Toolkit (as a way to bypass the "convert" part of the program) because (as above) doing so will cause other problems with tracking the records through the OAI Toolkit. If this process doesn't reveal the errors you could also view the file in a Hex Editor if you are familiar with Marc encoding conventions, such as field separators, etc.

Likely Errors

When debugging MARC records that fail to convert, pay special attention to the following potential problems which we have encountered:

1. Character Encoding Errors: Records that have special UTF-8 characters may cause problems, and you may want to verify that the input file encoding is correct for the records being loaded within that file. If it is not, this will throw off the reading of the records since everything is position-based. If this appears to be an issue, try separating records into two separate files by their encoding (UTF-8 and MARC-8) and making sure that the encoding of each file is identified correctly and set properly before converting the file.
2. Improperly formatted Leader: length indicator, bad character such as '?', mandatory field values missing. Note that some ILS's (such as Voyager) do not allow you to edit all positions of the Leader by using its cataloging module. In such cases, you may need to edit the record using MARCEdit or another program and then reload the corrected record back into the ILS.
3. Empty subfields, especially if they occur at the end of a field or at the end of a record. These should be deleted. (For the future, if you use cataloging templates in your ILS when creating or editing records, make sure that catalogers delete any subfields that don't contain data to avoid creating additional records with this problem!)
4. The MARC record itself may be corrupt, meaning that field and group separators are in the wrong place. This is definitely more complicated to detect, but could possibly be seen when opening a record in MARCEdit. Look especially for end-of-field terminators that occur within a field – one

solution would be to replace these with a space if found.

5. Illegal characters. These can be difficult to spot using either the ILS or MARCEdit. It is also possible to find them by viewing the raw MARC data (try using WordPad or a Hex editor). Check “Understanding MARC Bibliographic” <http://www.loc.gov/marc/umb> p. 21 for an explanation of how the Directory works. Illegal characters may show up as multiple “boxes” (characters that can’t be displayed properly) that appear adjacent to each other within the record. If you see multiple “boxes” next to each other within the record (except as the last two characters in the record), examine the record using either the ILS or another tool to see if you can spot and delete the illegal characters. (Note that it may be possible to delete these characters even if the tool you’re using doesn’t display them! Try hitting the delete key a few times to make sure there’s nothing lurking in the field!) Do NOT attempt to edit the raw MARC, because it may disrupt the character counts between the directory and the rest of the record.
6. Discrepancies between the directory and the rest of the MARC record. We have found some instances where records that failed had these problems – especially when multiple fields with the same tag have been reordered and the directory was not updated. It has been our experience that correcting other problem records in a file may make these records convert correctly, so it is unlikely that you will need to correct records with this kind of discrepancy. Try everything else first!

Using the XC OAI Toolkit Librarian Load Logs for MARC records

This log is likely to give a more accurate description of a problem within a specific record than the “convert” log. This log identifies both the record number and the type of record (bibliographic, holdings, etc.). While this log does not specifically identify the actual tag where the error is found, it provides a snippet of text from the record both before and after the error, which should make it possible to find the error either visually or by doing a “find” on the text of the record. The log also provides a marker in the text snippet right after the error occurs. We recommend correcting the records in the original repository (e.g. the ILS) whenever possible.

Likely Errors

1. Illegal or missing subfield codes and subfield names; subfield codes that are upper case instead of lower case.
2. Problems with the record Leader. Again, since some ILS’s do not allow you to edit all positions of the Leader you may need to edit the record using MARCEdit and then reload the corrected record back into the ILS.

Examples:

invalid character in XML

If we allow the weird characters in the conversion phase, this will cause an error at XML validation time:

```
ERROR - The MARC record marc.xml#3699979 isn't well formed. Please correct
the errors and load again. Cause:  has invalid XML character:  &#17 The
error is at line #1 char #133.
Source:
slim"><record><leader>02347cam a2200409 a 45&#17;0</leader><controlfield
-----^
The above arrow shows the area location where the error is.
```

This shows that there is a ASCII control character in MARC record's Leader.

invalid Leader

```
ERROR - The MARC record bibs_w_MFHDS01.xml#1919 of Type BIBLIOGRAPHIC
isn't well formed. Please correct the errors and load again. Cause:
content of record leader has non-valid value: '00000er>010000000a22003A'
The error is at line #1 char #139.
Source:
<record><leader>02401cas a2200625 | 4500</leader><controlfield
-----^
The above arrow shows the area location where the error is.
```

The record, which has record ID 26215, is not valid according to the MARCXML schema, because the Leader is different than the allowable values (which are described as a regular expression). The error is caught at the 1st line's 142nd character position of the record (not of the file).

invalid subfield code

```
ERROR - The MARC record bibs_w_MFHDS01.xml#5578 of Type HOLDINGS isn't
well formed. Please correct the errors and load again. Cause: subfield
code has non-valid value: 'V' The error is at line #1 char #871.
Source:
ubfield code="a">N</subfield><subfield code="V"/>
-----^
The above arrow shows the area location where the error is.
```

The record, which has record ID 76443, is not valid according to the MARCXML schema, because the subfield's code attribute is different than the allowable values (which are described as a regular expression). In this case the code's value is a space character, which is invalid. The error is caught at the 871st character position of the XML record (not of the file).

invalid indicator

```
ERROR- The MARC record marc.xml#1757639 of Type BIBLIGRAPHIC isn't well
formed. Please correct the errors and load again. Cause: 1st indicator of
data field has non-valid value: 'A' The error is at line #20 char #44.
```

Source:

```
datafield><datafield tag="740" ind1="0" ind2="A"><subfield code="a">
```

```
-----^
```

The above arrow shows the area location where the error is.

The record, which has record ID 1757639, is not valid according to the MARCXML schema, because the indicator attribute is different than the allowable values (which are described as a regular expression). The valid values are: any number between 0 and 9, any lower case ascii character or a space. In this case ind1 is "|" (pipeline character) which is not allowed. The error is caught at the 20th line's 44th character position of the record (not of the file).

malformed timestamp

```
The MARC record marc.xml#26244 has malformed timestamp:
'19900716085408.0 '. Converted to '19900716085408.0'
```

The record, which has record ID 26244, has a bad date representation value (see the space at the end), but it is corrected by the application, thus the record will be imported into the database and became harvestable.

Contacting Us

The suggestions for debugging errors in this document are based upon experiences at the University of Rochester and at Notre Dame (many thanks to Rick Johnson at Notre Dame for his input into this document!) As more users gain experience converting their MARC data using the XC OAI Toolkit and share their experiences with us, we will add them to this document. Please send any suggestions or questions about the Convert and Load processes – to Shreyansh Vakil at svakil@library.rochester.edu.