

## Appendix C: Batchload Validation for Bibliographic Records

Last revised September 2010

|     |  |    |
|-----|--|----|
| C.1 | Introduction.....  | 2  |
| C.2 | Timing of validation in batchload for bibliographic records .....                        | 3  |
| C.3 | Definitions of validation severity levels .....  | 4  |
| C.4 | Detailed error conditions for relaxed validation .....                                   | 6  |
| C.5 | Severity level 1 error messages.....   | 7  |
| C.6 | Severity level 2 error messages.....   | 8  |
| C.7 | Severity levels 3 and 4 error messages .....   | 10 |
| C.8 | List of all batchload validation error messages for bibliographic record processing .... | 11 |
| C.9 | Key to placeholder values in error messages .....  | 12 |

## C.1 Introduction

Batchload Error Validation is a software application used to automatically check the MARC 21 content designation and record format (structure) in bibliographic records that libraries send OCLC for batch processing. Validation can identify obvious errors for possible correction to improve matching and retrieval. “Quality” of the bibliographic records is not determined by validation.

### C.1.1 The scope of error validation

Error Validation does not check all MARC 21 content designators. Some MARC 21 rule violations are ignored because the error does not affect record retrieval (indexing), or represents local cataloging practices, or is too difficult to validate automatically.

For example, the following errors are not checked or reported because they do not affect record retrieval:

- Codes in subfield 2 in various fields and extraneous punctuation in ‡4 and ‡5
- Repeatability of subfields, except for ‡a and ‡b of field 245

### C.1.2 Validation via Connexion client or browser

The rules outlined in this chapter do **not** apply to record validation in OCLC Connexion. Record validation in Connexion is different from that used for batchload. For information on Connexion validation:

- See “Validate records online or offline” in Connexion client documentation at: [http://www.oclc.org/support/documentation/connexion/client/cataloging/editbib/#cat\\_edit\\_validate\\_records](http://www.oclc.org/support/documentation/connexion/client/cataloging/editbib/#cat_edit_validate_records)
- See “Validate bibliographic or authority record” in Connexion browser documentation at: [http://www.oclc.org/support/documentation/connexion/browser/cataloging/edit\\_bib\\_records/default.htm#wc-bib-af-validate](http://www.oclc.org/support/documentation/connexion/browser/cataloging/edit_bib_records/default.htm#wc-bib-af-validate)

## C.2 Timing of validation in batchload for bibliographic records

Batchload validation is used at two points during batchload: first, when a batchload specialist initially analyzes data sent by a library for processing, and second, during preprocessing, when records are evaluated for adding to WorldCat as original records.

### C.2.1 Validation for initial analysis of data

Batchload specialists use validation during initial analysis of record files to identify any critical errors. The batchload specialist runs the file through Error Validation software to identify the errors and then attempts to fix or mitigate the errors with processing routines that correct or normalize the data (that is, make data consistent). Validation is just one aspect of initial analysis.

### C.2.2 Validation for evaluating records before adding to WorldCat

During processing, every record processed in a file undergoes validation, and each error in a record is detected, recorded, and identified by severity level (severity levels are defined in the next section of this appendix). Once all errors are detected and assigned severity levels, the highest becomes the severity level of the record. If no errors are detected, the record error severity level is zero.

Records not matched with WorldCat records during batch processing are candidates for adding to WorldCat as original records, if requested by the library. The severity level of each record is checked, and records are either added to WorldCat or placed in an “unresolved record” file, based on the rules documented in this appendix. For information on unresolved record files, see chapter 5 “Exception Data (Records Remaining).”

Error severity levels have no effect on matched records used to set or delete holdings.

## C.3 Definitions of validation severity levels

In current practice, errors are classified by severity levels:

**Correctable errors** are incorrect indicator or coded values in situations where a correct or acceptable value can be supplied by default. Examples are:

- Non-blank indicator values where the indicator is undefined
- Some cases of blanks in fixed-field positions where blanks are not defined as valid coded values.

Invalid codes are replaced with default values and the record is counted as corrected unless it also has a more severe error.

**Severity level 1 errors** are invalid characters or content designators that are regarded as acceptable. The invalid character is passed through unchanged or replaced with an acceptable substitute. These errors are included in validation error reports (see appendix E for more about reports). The record is counted as containing minor errors unless the record also contains a more severe error.

**Severity level 2 errors** include most cases of invalid characters or content designators. For errors at this level, an attempt is made to correct them or they are simply identified as major. In most cases, the invalid element is replaced with a fill character or other acceptable substitute.

Records are counted as having major errors but are passed along for matching in WorldCat if they have no higher level errors. If they are unmatched, whether the records are added to WorldCat or not depends on the rules outlined in this appendix.

**Severity level 3 errors** are those which preclude successful processing. Examples include:

- Missing 245 field
- Invalid Record Status code
- 008 field that is too short.

Records with these errors are passed along for matching in WorldCat but are never added. These errors are reported, the record is counted as “rejected,” and it is not passed on to the next processing step.

**Severity Level 4 errors** are errors in the MARC record structure itself. Such errors prevent any further processing, including a formatted record display. These errors are included in the validation error report and counted as rejected.

All errors in each record are reported in a validation error report.

### C.3.1 Adding records to WorldCat: Strict versus relaxed validation with “conditional adds” option

Historically, batchload did not have the concept of major and minor errors. All errors were treated as Severity Level 2, which is the equivalent of today's "strict" validation. If any of the recognized errors appeared in a record, the record was not added to WorldCat.

Currently, batchload validation has been “relaxed.” Relaxed validation categorizes the severity of errors and then corrects some errors and allows records with minor errors to be added to WorldCat.

A batchload specialist may still choose to use strict validation, but by default, most batchload projects are now processed using relaxed validation in combination with a processing option called “conditional adds.” Strict validation is always used for National Library processing.

Combining conditional adds and relaxed validation for most projects is a way of trying to strike a balance between getting records into WorldCat and maintaining WorldCat record quality. The rules in this appendix describe this default combination.

**Rare exception:** Batchload specialists can set a processing option to “unconditional adds,” which allows adding all but the most problematic records to WorldCat. The rules for unconditional adds are rarely used and are considered only for unusual projects. This exception is beyond the scope of this document.

## C.4 Detailed error conditions for relaxed validation

The following list explains how specific error conditions are handled under relaxed validation.

- **Invalid indicator values:** If only one value is defined for the indicator, it is set to that value and no error is reported. Otherwise the invalid value is replaced by an error fill character. The severity level is 1. Applies to all variable data fields.
- **Invalid subfield codes:** If the existing subfield code is in the ranges *a-z*, *0-5*, or *7-9*, it is left unchanged and the severity level is 1. If the existing value is *6*, the severity level is 2. Other subfield code values are replaced with a fill character, and the severity level is 2. Applies to all variable data fields.
- **Multiple occurrences of non-repeatable fields:** The severity level is 1. Applies to all variable data fields except 010, 029, and 245.
- **Invalid code in ModRec:** The invalid code is changed to a blank. No error is reported.
- **Redundant escape sequence:** No error is reported.
- **Mutually exclusive fields:** No action is taken. The severity level is 1.
- **Fixed field too short:** The field is treated as if it were padded with fill characters to the minimum acceptable length. The severity level is 1. Applies to fields 006 and 007.
- **Fixed field too long:** The field is truncated to an acceptable length. No error is reported. Applies to fields 006-008.
- **Fixed field format-specific elements (008/18-34 and 006/01-17):** The MARC fill character is considered valid for non-critical elements. Critical elements are those involved in record type determination and matching, such as Type of Material. Errors in critical elements remain severity level 2. For non-critical elements, if the existing erroneous value is a valid ASCII graphic, the value is left unchanged. Otherwise, the value is replaced with an error fill character. The severity level is 1.
- **Invalid tag:** If the tag is non-numeric or of the form 00X, or if it is marked “not legal for input,” the severity level remains 2. Otherwise, the tag is not altered, and the contents of the field are validated according to the generic field rules—indicator values of blank and *0-9* are valid; subfield codes *a-z* and *0-9* are valid). The severity level is 1 unless the field contains a more severe error.
- **Field 007:** The MARC fill character is considered valid for use in positions 03 and beyond. Position 02 is always forced to blank. For positions 03 and beyond, if the existing erroneous value is alphabetic, numeric, a blank, or a hyphen, the value is retained. Otherwise, the value is replaced with an error fill character. The severity level is 1.

## C.5 Severity level 1 error messages

**Note:** See section C.9 for definitions of placeholder values—such as *ttt*, *nn* or *nnn*, *xx*—in error messages.

### ttt FIELD INCORRECT LENGTH

Field 006 or 007 has other than the defined length, or field 008 is too long. Characters missing from field 006 are treated as if fill characters were present. Missing characters in field 007 are ignored. Excess characters in all fields are ignored.

### FIELD ttt MAY NOT BE REPEATED

A field designated as non-repeatable but not marked absolutely non-repeatable occurs more than once in the record. Example: **240**.

### FIELD ttt MAY NOT OCCUR WITH FIELD ttt

Two fields designated as mutually exclusive appear in the same record. Example: **100** with **130**.

### INCORRECT VALUE 'c' (ASCII X'xx') IN POSITION nnn (007 FIELD POSITION nn)

A single-character coded value in field 007 is not valid.

### INVALID DATA IN POSITION nn OF ttt FIELD (element)

A single-character coded value in field 006 or 008 is not valid. The element is not critical for matching and indexing.

### INVALID DATA IN POSITIONS nn-nn OF ttt FIELD (element)

A multi-character coded value in field 006 or 008 is not valid. The element is not critical for matching and indexing.

### INVALID DATA IN POSITIONS nnn-nnn (007 FIELD POSITIONS nn-nn)

A multi-character coded value in field 007 is not valid. The erroneous value is replaced with error fill characters.

### INVALID DESCRIPTIVE CATALOGING FORM CODE 'c' (ASCII X'xx')

The Descriptive Cataloging Form code (leader position 18) is not valid. The erroneous value is replaced with an error fill character.

### INVALID FIRST INDICATOR 'c' (ASCII X'xx') IN POSITION nnn (ttt FIELD)

The value of the first indicator is not valid in context. It is replaced with an error fill character.

### INVALID SECOND INDICATOR 'c' (ASCII X'xx') IN POSITION nnn (ttt FIELD)

The value of the second indicator is not valid in context. It is replaced with an error fill character.

### INVALID SUBFIELD CODE 'c' (ASCII X'xx') IN POSITION nnn (ttt FIELD)

The subfield code is not valid in context.

### INVALID TAG 'ttt' (FIELD BEGINNING IN POSITION nnn)

The tag is numeric, not defined, and not a control field.

### REDUNDANT ESCAPE SEQUENCE IN POSITION nnn (ttt FIELD)

An escape sequence appears when the character set it designates is already in effect.

## C.6 Severity level 2 error messages

**Note:** See section C.9 for definitions of placeholder values—such as *ttt*, *nn* or *nnn*, *xx*—in error messages.

### BIB RECORD EXCEEDS MAXIMUM LENGTH - RECORD TRUNCATED

The total length of the record when converted to internal processing format exceeds the system limit. Occurrence of this error is extremely rare since the system limit was raised to 32,000 bytes.

### FIELD TERMINATOR MISSING IN POSITION *nnn* (*ttt* FIELD)

The final position of a field as indicated by the length in its directory entry does not contain a field terminator. A terminator is stored into the final position.

### FIELD *ttt* MAY NOT BE REPEATED

A field designated as absolutely non-repeatable occurs more than once in the record.  
Example: **245**.

### ILLEGAL CHARACTER X'*xx*' (ASCII X'*xx*') IN POSITION *nnn* (*ttt* FIELD)

The reported position contains a character which is not a valid MARC character or which OCLC does not yet support. The invalid character is replaced with an error fill character.

### ILLEGAL ESCAPE SEQUENCE IN POSITION *nnn* (*ttt* FIELD)

The characters following an escape do not constitute a recognized character set identifier and the standard character set is currently in effect. The escape character is replaced with an error fill character.

### INCORRECT DATA IN 042 FIELD

The content of field 042 subfield a does not precisely match one of the known valid codes. For Standard and Group processing, most 042 fields are deleted prior to validation.

### INVALID DATA IN POSITION *nn* OF *ttt* FIELD (element)

A single-character coded value in field 006 or 008 is not valid. The element is critical for matching and indexing.

### INVALID DATA IN POSITIONS *nn-nn* OF *ttt* FIELD (element)

A multi-character coded value in field 006 or 008 is not valid. The element is critical for matching and indexing.

### INVALID EACC CHARACTER *eeeeee* IN POSITION *nnn* (*ttt* FIELD)

The reported three-byte sequence occurs within an EACC string but is not a valid EACC character. The data is not changed.

### INVALID ENCODING LEVEL '*c*' (ASCII X'*xx*')

The Encoding Level value (leader position 17) is not valid. The erroneous value is replaced with an error fill character. This error does not occur except when there is an error in the setup.

### INVALID SUBFIELD CODE '*c*' (ASCII X'*xx*') IN POSITION *nnn* (*ttt* FIELD)

The subfield code is not a letter or number. The erroneous code is replaced with an error fill character.

INVALID TAG 'ttt' (FIELD BEGINNING IN POSITION nnn)

The tag is non-numeric, not valid for input, or is restricted from use in the category of records being processed.

NO ROOM IN RECORD FOR 040 FIELD

There is not enough room in the internal processing record for the 040 field, which is constructed after conversion. Occurrences of this error are extremely rare since the system limit was raised to 32,000 bytes.

NON-STANDARD CODED CHARACTER IN POSITION nnn (ttt FIELD)

A variant numeric character reference, as produced by the UTF-8 converter, is present at the indicated position. These strings have the form {uXXXX}, where "XXXX" is the Unicode value of the character in hexadecimal.

SUBFIELD \$6 PRESENT WITHOUT FIELD 880

Subfield 6 was found in one or more fields, but no field 880 is present in the record.

SUBFIELD \$% MAY NOT BE REPEATED IN FIELD ttt

"%" is "a" or "b". "ttt" is 245.

SUBFIELD DELIMITER IS NOT FIRST IN POSITION nnn (ttt FIELD)

The first position following the indicators in a variable data field is not a subfield delimiter. The initial characters of this field will be treated as an implicit subfield a.

## C.7 Severity levels 3 and 4 error messages

### Notes:

- Level 4 errors in the messages listed below are marked with [4] at the end of the message description.
- See section C.9 for definitions of placeholder values—such as *ttt*, *nn* or *nnn*, *xx*—in messages.

#### 001 FIELD MISSING OR EMPTY

Batchload requires a non-blank record identifier in field 001.

#### 008 FIELD INCORRECT LENGTH

The 008 field is shorter than 41 bytes. If the field is too long, the error is severity level 2.

#### FIELD TERMINATOR MISSING IN POSITION *nnn*

A field terminator was not found in the last byte of the record directory as indicated by the Base Address of Data. [4]

#### FIELD *ttt* REQUIRED BUT NOT PRESENT

The tag is 008 or 245. Both are mandatory.

#### ILLEGAL ESCAPE SEQUENCE IN POSITION *nnn* (*ttt* FIELD)

The characters following an escape do not constitute a recognized character set identifier and the standard character set is not in effect. This error is severity level 2 if the standard character set is currently in effect.

#### INVALID BASE ADDRESS

The Base Address of Data (leader positions 12-16) does not contain a valid value. The value must be one greater than a multiple of 12. [4]

#### INVALID RECORD STATUS CODE

The record status code is not a recognized valid value; therefore, the transaction is undefined.

#### INVALID RECORD TYPE/BIB LEVEL

The Record Type or Bibliographic Level code, or combination of the two, is invalid. Thus the record format is undefined and validation cannot be performed.

#### NO VALID HOLDINGS SYMBOL FOUND

The record lacks a valid library identifier (field 949 subfield I) during group processing. Thus no holdings can be set.

#### NON-NUMERIC X'xx' (ASCII X'xx') IN POSITION *nnn*

The Base Address of Data (leader positions 12-16) or any of the first seven positions of a directory entry contains a non-numeric character. [4]

#### SUBFIELD \$6 NOT FIRST IN FIELD - POSITION *nnn* (*ttt* FIELD)

Subfield 6 must be the first subfield in any field other than 886 or 887.

#### SUBFIELD \$6 STRUCTURE INVALID - POSITION *nnn* (*ttt* FIELD)

The subfield does not have the required structure.

## C.8 List of all batchload validation error messages for bibliographic record processing

**Note:** See section C.9 for definitions of placeholder values—such as *t*, *nn* or *nnn*, *xx*—in error messages.

| Error Number | Error Message Text   | Severity Level |
|--------------|--|----------------|
| 01           | INVALID ENCODING LEVEL 'c' (ASCII X'xx')   | 2              |
| 02           | INVALID DESCRIPTIVE CATALOGING FORM CODE 'c' (ASCII X'xx')                         | 1              |
| 03           | INVALID RECORD TYPE/BIB LEVEL  | 3              |
| 04           | INVALID RECORD STATUS CODE   | 3              |
| 05           | SUBFIELD DELIMITER IS NOT FIRST IN POSITION nnn (t <sub>tt</sub> FIELD)            | 2              |
| 06           | INVALID TAG 't <sub>tt</sub> ' (FIELD BEGINNING IN POSITION nnn)                   | 1 or 2         |
| 07           | INVALID FIRST INDICATOR 'c' (ASCII X'xx') IN POSITION nnn (t <sub>tt</sub> FIELD)  | 1              |
| 08           | INVALID SECOND INDICATOR 'c' (ASCII X'xx') IN POSITION nnn (t <sub>tt</sub> FIELD) | 1              |
| 09           | ILLEGAL CHARACTER X'xx' (ASCII X'xx') IN POSITION nnn (t <sub>tt</sub> FIELD)      | 2              |
| 10           | INVALID SUBFIELD CODE 'c' (ASCII X'xx') IN POSITION nnn (t <sub>tt</sub> FIELD)    | 1 or 2         |
| 11           | FIELD TERMINATOR MISSING IN POSITION nnn (t <sub>tt</sub> FIELD)                   | 2              |
| 12           | NON-NUMERIC X'xx' (ASCII X'xx') IN POSITION nnn                                    | 4              |
| 13           | INVALID BASE ADDRESS   | 4              |
| 14           | FIELD TERMINATOR MISSING IN POSITION nnn   | 4              |
| 15           | FIELD t <sub>tt</sub> MAY NOT BE REPEATED  | 1 or 2         |
| 16           | FIELD t <sub>tt</sub> MAY NOT OCCUR WITH FIELD t <sub>tt</sub>                     | 1              |
| 17           | FIELD t <sub>tt</sub> REQUIRED BUT NOT PRESENT                                     | 3              |
| 18           | INCORRECT VALUE 'c' (ASCII X'xx') IN POSITION nnn (007 FIELD POSITION nn)          | 1              |
| 20           | t <sub>tt</sub> FIELD INCORRECT LENGTH   | 1 or 3         |
| 21           | INVALID DATA IN POSITIONS nn-nn OF t <sub>tt</sub> FIELD (element)                 | 1 or 2         |
| 22           | INVALID DATA IN POSITION nn OF t <sub>tt</sub> FIELD (element)                     | 1 or 2         |
| 23           | ILLEGAL ESCAPE SEQUENCE IN POSITION nnn (t <sub>tt</sub> FIELD)                    | 2 or 3         |
| 24           | REDUNDANT ESCAPE SEQUENCE IN POSITION nnn (t <sub>tt</sub> FIELD)                  | 0 or 1         |
| 27           | INVALID DATA IN POSITIONS nnn- <i>nnn</i> (007 FIELD POSITIONS 9-10)               | 1              |
| 28           | SUBFIELD \$6 PRESENT WITHOUT FIELD 880   | 2              |
| 29           | SUBFIELD \$6 NOT FIRST IN FIELD - POSITION nnn (t <sub>tt</sub> FIELD)             | 3              |
| 30           | SUBFIELD \$6 STRUCTURE INVALID - POSITION nnn (t <sub>tt</sub> FIELD)              | 3              |
| 31           | INVALID DATA IN POSITIONS nnn- <i>nnn</i> (007 FIELD POSITIONS 6-8)                | 1              |
| 32           | INVALID EACC CHARACTER eeeee IN POSITION nnn (t <sub>tt</sub> FIELD)               | 2              |
| 33           | INCORRECT DATA IN 042 FIELD  | 2              |
| 34           | NON-STANDARD CODED CHARACTER IN POSITION nnn (t <sub>tt</sub> FIELD)               | 2              |
| 35           | SUBFIELD \$? MAY NOT BE REPEATED IN FIELD t <sub>tt</sub>                          | 2              |
| 52           | BIB RECORD EXCEEDS MAXIMUM LENGTH - RECORD TRUNCATED                               | 2              |
| 53           | 001 FIELD MISSING OR EMPTY   | 3              |
| 56           | NO ROOM IN RECORD FOR 040 FIELD  | 2              |
| 57           | NO VALID HOLDINGS SYMBOL FOUND   | 3              |

## C.9 Key to placeholder values in error messages

Parts of the text of the error messages in this appendix consist of placeholders for actual values, which are inserted when records are validated. The placeholders are:

| <b>Placeholder</b>      | <b>Definition</b>   |
|-------------------------|---|
| <b>c</b>                | A character, presented in single quotes: for example, 'E'. If the character is not printable, the hex value is displayed instead. |
| <b>eeeeee</b>           | An EACC character, presented as a string of six hexadecimal digits: for example, <b>21344B</b> .                                  |
| <b>element</b>          | One of the mnemonic labels given to elements of the 006 and 008 fields in online displays.  |
| <b>nn</b> or <b>nnn</b> | A decimal offset from the beginning of a record or field. Also used to identify elements of the fixed fields by position.         |
| <b>rrrr</b>             | An RLIN library identifier: for example, <b>NYPG</b> .  |
| <b>ttt</b>              | A three-digit tag: for example, <b>245</b> .  |
| <b>xx</b>               | The hexadecimal value of an ASCII or EBDIC character, usually displayed in the form X'xx'.  |
| <b>?</b>                | A single lowercase letter or digit.   |