Thank you for joining us!
We will begin at 1:00 PM Eastern Daylight Time

Attendees are in listen-only mode

AskQC@oclc.org (Ask Quality Control) is the longstanding email address to which catalogers can send questions to OCLC Metadata Quality staff about cataloging policies, standards, and practices.
Virtual AskQC Office Hours
Bibliographic Record Validation

OCLC Metadata Quality
Housekeeping

• Please remain muted
• Today’s session is being recorded
• Recordings, slides, and notes are available at https://oc.lc/askqc
On the call today

Robert Bremer
Senior Consulting Database Specialist

Cynthia Whitacre
Manager, Metadata Policy

Jay Weitz
Senior Consulting Database Specialist

Patty Treboni
Database Specialist II

Bryan Baldus
Consulting Database Specialist

Nathan Putnam
Director, Metadata Quality
BIBLIOGRAPHIC RECORD VALIDATION
Validation of Bibliographic Records

- Use of validation helps to ensure that bibliographic records contain valid MARC coding to facilitate processing and use.
- Current validation continues what was initially developed for the old PRISM service—the predecessor to Connexion.
- Validation is a separate service that can be called by various applications, e.g., Data Ingest, Connexion, Record Manager.
Validation of Bibliographic Records

- Process uses a database of MARC data elements, i.e., fixed-field elements, tags, indicators, and subfields
- Updated annually based on published MARC updates as well as several times a year to add newly announced MARC codes
- Validation does not change information, but rather only reports back errors—a reformat of the record occurs when validating online
Validation Database Contents

- Information on fixed-field elements, tags, indicators, subfields, including data on:
  - Repeatability
  - Length (minimum and maximum)
  - Type of data allowed
  - Lists of codes (which can be shared between elements, e.g., country, language, geographic area codes, relators, etc.)
  - Information on relationships between elements
  - References to special rules used in processing
Validation Processing – Fixed Field

• Checks code validity
• Example:

DtSt (008/06) - invalid code
DtSt: \textcolor{red}{z} Dates: 1963,____
Validation Processing – Fixed Field

• Checks code repeatability (for elements with multiple positions)
• Example:

Ills (008/18-21) - not allowed to repeat codes
Ills: ab\_b
Validation Processing – Fixed Field

• Checks code spacing (for elements with multiple positions)
• Example:

  Cont (008/24-27) - spaces not allowed between codes
  Cont: b c

  Note: you are unlikely to see this error in the Connexion client
Validation Processing – Field

- Checks tag validity
- Example:

206 - invalid tag

Validation Processing – Field

• Checks minimum/maximum occurrence
• Example:

245 - occurs too many times
245 04 The Rand McNally book of favorite pastimes / ṭc illustrated by Dorothy Grider.
245 00 Favorite pastimes
Validation Processing – Field

• Checks length (Leader and control fields only, i.e., 005, 006, 008)

• Example:

006 occurrence 1 - too long
006  a  000 0 0

Note: you are unlikely to see these kinds of errors
Validation Processing – Indicator

• Checks code validity
• Example:

100 occurrence 1, indicator 1 - invalid code
100 occurrence 1, indicator 2 - invalid code
100 55 Shakespeare, William, ‡d 1564-1616.
Validation Processing – Subfield

• Checks subfield validity
• Example:

245 occurrence 1 - invalid subfield $x$

245 10  Dracula / $x$ by Bram Stoker.
Validation Processing – Subfield

- Checks minimum/maximum occurrence
- Example:

  700 occurrence 1, $d - occurs too many times
  700 1  Twain, Mark, ‡d 1835-1910 ‡d (Spirit)
Validation Processing – Subfield

• Checks minimum/maximum length

• Examples:

  022 occurrence 1, $a occurrence 1 - too short
  022  00029769

  022 occurrence 1, $a occurrence 1 - too long
  022  0002-97669
Validation Processing – Subfield

- Checks code validity (for subfields containing codes)
- Code lists are shared between fields, e.g., 600 $2, 610 $2, 650 $2, etc., use the same subject source code list
- Example:

  043 occurrence 1, $a occurrence 1 - invalid code
  043 e-uk-sc
Validation Processing – Subfield

• Checks data validity (for subfields not containing codes)
• Data can be specified as Latin, ASCII, numeric, Unicode, etc.
• Example:

  306 occurrence 1, $a occurrence 1, position 2 - invalid character - data must be numeric characters

  306 0|1500
Validation Processing – Non-Latin Data

• Non-Latin script data is retained in 880 fields
• Corresponding tag is in $6 and is what displays to catalogers
• Tag in $6 governs which rules are used to check the field
• Any Latin data types associated with the non-880 tag are automatically processed as Unicode allowing for non-Latin script

880 10 ‡6 245-02 ‡a Воспоминания
Validation Processing – Relationships

- Relationships can be made for all types of elements, i.e., fixed field, field, indicator, and subfield
- Relationships can be made from one type of element to another or to the same type, e.g.,

  255 occurrence 1 - invalid relationship - when element is present, then Type (Leader/06) must not be equal to c

  DtSt (008/06) - invalid relationship - when element is equal to m, then Date2 (008/11-14) must not be equal to blank
Validation Processing – Relationships

• Relationships fall into three types: Must, Can’t, and Any
• Must – if an element is present or equal to some value, then another element MUST be present or equal to some value
• Can’t – if an element is present or equal to some value, then another element CAN’T be present or equal to some value
• Any – if an element is present or equal to some value, then ANY other element from a set of possible elements must be present or equal to some value
Validation Processing – Relationship Examples

- **Must** – *Type (Leader/06) - invalid relationship - when element is equal to 0, then TMat (008/33) must equal b*
- **Can’t** – *110 occurrence 1 - invalid relationship - when element is present, then 100 must not be present*
- **Any** – *240 occurrence 1 - invalid relationship - when element is present, then 100, 110 or 111 must be present*
Validation Processing – Special Rules

• Special rules exist for situations falling outside routine checks against validation database:
  • Calculation of check digits in ISBNs, ISSNs, etc.
  • Checks structure of numbers in 024 based on indicators
  • Checks terms in 336-338 $a based on coding of $2
  • Checks structure of numbers in 760-787 $w
  • Checks structure of numbers or URIs in $0
  • Checks codes or structure of URIs in $4
  • Checks structure of URLs in 856 $u
  • Performs record level check, i.e., must have 008, 245 $a/$k, etc.
Validation has three levels, i.e., Full, Basic, and Structure

- Full validation includes checking all elements and relationships
- Basic validation includes checking all elements except relationships
- Structure validation includes checking for existence of 008, 245, etc.

- Full validation occurs when using the validate command or when adding or replacing a record
- Basic and Structure validation exist as options for setting holdings and exporting records
Validation does not check subfield order

Most pre-existing errors do not need to be fixed when replacing a record

Edit restrictions exist separately from validation, but are invoked when validation occurs

Error messages are generated in a structured format which allowing them to be inserted into error message templates in different languages for displays and reports
Validation Processing – Non-Bibliographic Records

• Validation service is also used for other types of MARC-based records
• Different validation rule sets exist for processing LC name authority records, LC subject authority records, local bibliographic data, and local holdings records.
Questions about validation?

Please submit questions through chat
Any general questions?

Please submit questions through chat
Thank you!

Send cataloging policy questions at anytime to: askqc@oclc.org

Session links available at: https://oc.lc/askqc

Next Virtual AskQC Office Hours:
Wednesday, June 27, 2018
1:00 PM Eastern Daylight Time

Topic: Expert Community