MARC field protections

- Background
- Current constraints of MARC Field Protections:
  - Repeatable and Non-repeatable fields:
  - Sample Field Protection Settings
  - Field Protection Examples and Expected Actions
  - MARC field protection scenarios
  - Overriding Field Protections

Background

- First iteration created in MODDICONV-137
- Bugfixes required and will be delivered in Lotus, so this page has been created to document expected behavior for 1) Protecting MARC field and then 2) Overriding MARC field protections. (overrides not yet documented here)
- Related bugs
  - https://issues.folio.org/browse/MODDATAIMP-643
  - https://issues.folio.org/browse/MODDICORE-248
  - https://issues.folio.org/browse/MODDICORE-146

Current constraints of MARC Field Protections:

1. MARC field protections are only invoked if a MARC Update action is included in the job profile.
   a. Based on feedback from the community, consider changing the behavior so that the field protections are invoked automatically/implicitly.
2. When evaluating data in a subfield, MARC field protections are not case-sensitive. For example $a NcD, $a NCD, $a Ncd, $a ncd, $a ncD would all be considered equivalent and duplicates of each other.
   a. After testing and early usage, if the community decides that the field protections must take upper and lower case into account for data in subfields, consider changing the behavior so that the field protections are case-sensitive rather than case-insensitive.
3. For data in a subfield, the field protection is based on the entire data string.
   a. Consider supporting wildcards, begins, ends, contains in the future.
4. One master list of MARC field protections is maintained in Settings/Data import.
   a. Per conversation with the MARC Holdings libraries, MARC field protections will NOT be applied to MARC Holdings records.
   b. Unless/until a future use case is identified that necessitates differentiation, this same list of field protections will be used for MARC Bibliographic and MARC Authority records.
5. MARC field protections are controlled at the tenant level, though field protections can be overridden for a particular action profile/job profile.
   a. Based on feedback from the community, consider adding the ability to include additional field protections in individual job profiles, for MARC fields that are used infrequently or do not normally need to be protected.
6. Question for librarians: can we assume that LDR, 002-009 fields do not need field protection? Per MM SIG and DI subgroup, March 2022: fields 006 and 007 should allow field protection, but none of the other control fields (LDR, 002, 003, 004, 005, 008, 009).

Repeatable and Non-repeatable fields:

MARC field protections work differently for repeatable and non-repeatable fields. Here is a list of those fields (spanning all three MARC record types):

- Repeatable: all other MARC fields

Sample Field Protection Settings

<table>
<thead>
<tr>
<th>Field</th>
<th>Ind 1</th>
<th>Ind 2</th>
<th>Subfield</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>5</td>
<td>NcD</td>
</tr>
<tr>
<td>035</td>
<td>*</td>
<td>*</td>
<td>b</td>
<td>MiAaHDL</td>
</tr>
<tr>
<td>050</td>
<td>9</td>
<td>7</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>590</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>982</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Field Protection Examples and Expected Actions

<table>
<thead>
<tr>
<th>Field</th>
<th>Ind 1</th>
<th>Ind 2</th>
<th>Subfield</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>5</td>
<td>NcD</td>
</tr>
</tbody>
</table>

Add behavior for 006 and 007

Behavior for repeatable fields: Retain existing field. If incoming record has one or more different fields (varying by field, indicator 1 or 2, subfield, or data), add the incoming field(s); if incoming record has a field that is an exact match to an existing field, do not add a duplicate field (if possible).
Behavior for **non-repeatable** fields: Retain existing field. Discard incoming field.

- **Example 1**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming: 950 1 2 $a 325168 $5 NcD
  - Retain existing 950; add incoming 950 (since the $a data is different)

- **Example 2**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming: 950 3 4 $a 325168
  - Retain existing 950; add incoming 950 (since the indicators and the $a value are different)

- **Example 3**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming: 950 1 2 $a 325169 $5 NcD
  - Replace the existing 950 with the incoming 950 (which would be protected in future updates of the record)

- **Example 4**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming:
    - 950 1 2 $a 325167 $5 NcD
    - 950 1 2 $a 325167 $5 NcA
    - 950 1 2 $a 325168
    - 950 1 2 $a 325169 $5 NcD
  - Retain existing 950; add all incoming 950s except the first one (since the first one duplicates an existing field, and the rest do not)

- **Example 5**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming:
    - 950 1 2 $a 325167 $5 NcD
    - 950 1 2 $a 325167 $5 NcD
    - 950 1 2 $a 325168
    - 950 1 2 $a 325169 $5 NcD
  - Retain existing 950; add all incoming 950s except the first and second ones (since they duplicate an existing field, and the rest do not)

- **Example 6**
  - Existing: 950 1 2 $a 325167 $5 NcD
  - Incoming: no 950
  - Retain existing 950

- **Example 7**
  - Existing: 010 $a 12345678 $5 NcD
  - Incoming: 010 $a 657453647 $5 NcD
  - Retain existing 010; discard incoming 010 (since 010 is a non-repeatable field)

- **Example 8**
  - Existing field: 010 $a 12345678
  - Incoming: 010 $a 12345678 $5 NcD
  - Replace existing 010 with the incoming 010 (since 010 is a non-repeatable field, but the existing one is not protected)

- **Example 9**
  - Existing field: 010 $a 12345678
  - Incoming: no 010 field
  - Existing 010 will be discarded when SRS record is replaced, since it was not protected. Updated SRS record will not have an 010

<table>
<thead>
<tr>
<th>Field</th>
<th>Ind 1</th>
<th>Ind 2</th>
<th>Subfield</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>035</td>
<td>*</td>
<td>*</td>
<td>b</td>
<td>MiAaHDL</td>
</tr>
</tbody>
</table>

Behavior for **repeatable** fields: Retain existing 035. If incoming record has one or more different 035s (varying by indicator 1 or 2, subfield, or data), add the incoming 035(s); if incoming record has an 035 that is an exact match to an existing 035, do not add a duplicate 035.

Behavior for **non-repeatable** fields: Retain existing field. Discard incoming field.

035 is a repeatable field.

- **Example 1**
  - Existing: 035 12 $a 12345 $b MiAaHDL
  - Incoming: 035 34 $a 4327842 $b MiAaHDL
  - Retain existing 035; add incoming 035 (since the $a value is different)

- **Example 2**
  - Existing: 035 $a 12345
  - Incoming: 035 $a 4327842 $b MiAaHDL
  - Discard existing 035 (since it is not protected); add incoming 035

- **Example 3**
  - Existing: 035 $a 12345 $b MiAaHDL
  - Incoming: 035 $a 12345 $b MiBaHDL
  - Retain existing 035; add incoming 035 (since the $b value is different)

- **Example 4**
- **Existing**: 035 $a 12345 $b MiAaHDL
- **Incoming**:
  - 035 $a 12345 $b MiAaHDL
  - 035 $a 12345
  - 035 92 $a 6582634 $b MiAaHDL
  - 035 33 $a (OCoLC)743256132
- Retain existing 035; discard first incoming 035 (since it is a duplicate); add all other incoming 035s

**Example 5**

- **Existing**:
  - 035 $a 12345 $b MiAaHDL
  - 035 $a 12345
  - 035 $a 6582634 $b MiAaHDL
  - 035 $a (OCoLC)743256132
- **Incoming**:
  - 035 $a 6582636 $b MiAaHDL
  - 035 $a (OCoLC)467815642
- Retain the existing 035s with $b MiAaHDL; discard existing 035s without $b MiAaHDL (since they are not protected); add all incoming 035s.

**Example 6**

- **Existing**:
  - 035 $a 12345 $b MiAaHDL
  - 035 $a 12345
  - 035 $a 6582634 $b MiAaHDL
  - 035 $a (OCoLC)743256132
- **Incoming**: no 035s
- Retain the existing 035s with $b MiAaHDL; discard existing 035s without $b MiAaHDL (since they are not protected)

<table>
<thead>
<tr>
<th>Field</th>
<th>Ind 1</th>
<th>Ind 2</th>
<th>Subfield</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>050</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>MT123 $T68</td>
</tr>
</tbody>
</table>

Behavior for **repeatable** fields: Retain existing 050. If incoming record has one or more different 050s (varying by indicators 1 or 2, subfield, or data in the subfield), add the incoming 050(s); if incoming record has an 050 that is an exact match to an existing 050, do not add a duplicate 050 (if possible).

Behavior for **non-repeatable** fields: Retain existing field. Discard incoming field.

050 is a repeatable field.

**Example 1**

- **Existing**: 050 8 6 $a MT123 $b .T68 2021
- **Incoming**: 050 8 6 $a MT123 $b .T68 2021
- Discard existing 050 (since indicators 8 6 are not protected); add incoming 050

**Example 2**

- **Existing**: 050 9 7 $a MT123 $b .T68 2021
- **Incoming**: 050 8 6 $a MT123 $b .T68 2021
- Retain existing 050; add incoming 050 (since it is not an exact match to existing 050)

**Example 3**

- **Existing**: 050 9 7 $a MT123 $b .T68 2021
- **Incoming**: 050 9 7 $a MT123 $b .T68 2022
- Retain existing 050; add incoming 050 (since $b data is different)

**Example 4**

- **Existing field**: 050 9 7 $a MT123 $b .T68 2021
- **Incoming**: 090 9 7 $a MT123 $b .T68 2021
- Retain existing 050; add incoming 090

**Example 5**

- **Existing**: 050 9 7 $a MT123 $b .T68 2021
- 050 0 1 $a MT456
- **Incoming**: 050 0 1 $a MT456 $b .T68 2021
- 050 9 7 $a MT123 $b .T68 2021
- 090 0 1 $a MT456 $b .T68 2021
- Retain existing 050 with indicators 9 7; discard existing 050 with indicators 0 1; add incoming 050 with indicators 0 1; discard incoming 050 with indicators 9 7 (since it duplicates an existing 050); add incoming 090

**Example 6**

- **Existing**:
  - 050 9 7 $a MT123 $b .T68 2021
  - 050 0 1 $a MT456 $b .T68 2021
  - 090 0 1 $a MT456 $b .T68 2021
- **Incoming**: no 050 or 090
- Retain existing 050 with indicators 9 7; discard existing 050 with indicators 0 1; discard existing 090
Behavior for **repeatable** fields: Retain existing 590. If incoming record has one or more 590s (varying by indicators 1 or 2, subfield, or data in the subfield), add the incoming field(s); if incoming record has a 590 that is an exact match to an existing 590, do not add a duplicate 590 (if possible).

Behavior for **non-repeatable** fields: Retain existing field. Discard incoming field.

590 is a repeatable field.

- **Example 1**
  - Existing: 590 1 1 $a Some important note
  - Incoming: 590 2 2 $a Another important note
  - Retain existing 590; add incoming 590
  - Existing 590s would always be retained, and incoming 590s would always be added, since the indicators, subfields, and data do not matter. An incoming 590 would only be discarded if it duplicates an existing 590.

- **Example 2**
  - Existing:
    - 590 1 1 $a Some important note
    - 590 2 2 $a Another important note
  - Incoming: 590 1 1 $b Some other kind of note
  - Retain all existing 590s; add incoming 590

- **Example 3**
  - Existing:
    - 590 1 1 $a Some important note
    - 590 2 2 $a Another important note
    - 590 1 1 $b Some other kind of note
  - Incoming:
    - 590 1 1 $a Some important note
    - 590 2 1 $c Some important note
    - 590 1 1 $d Some important note
  - Retain all existing 590s; discard first incoming 590 (since it duplicates an existing one); add all other incoming 590s

- **Example 4**
  - Existing:
    - 590 1 1 $a Some important note
    - 590 2 2 $a Another important note
    - 590 1 1 $b Some other kind of note
    - 590 2 1 $c Some important note
    - 590 1 1 $d Some important note
    - 590 1 1 $a Some important note
  - Incoming: no 590s
  - Retain all existing 590s; discard the existing 591 (since it is not protected)

<table>
<thead>
<tr>
<th>Field</th>
<th>Ind 1</th>
<th>Ind 2</th>
<th>Subfield</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>982</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Behavior for **repeatable** fields: Retain existing 982. If incoming record has one or more different 982s (varying by indicators 1 or 2, subfield, or data in the subfield), add the incoming 982(s); if incoming record has a 982 that is an exact match to an existing 982, do not add a duplicate 982 (if possible).

Behavior for **non-repeatable** fields: Retain existing field. Discard incoming field.

982 is a repeatable field.

- **Example 1**
  - Existing: 982 $a 20210531
  - Incoming: 982 $a 20210531 $a 20210602
  - Retain existing 982; add incoming 982

- **Example 2**
  - Existing: 982 $a 20210531 $a 20210602
  - Incoming: 982 1 1 $a 20210531 $a 20210602
  - Retain existing 982; add incoming 982 (since indicators are different)

- **Example 3**
  - Existing: 982 $a 20210531 $a 20210602
  - Incoming:
    - 982 $a 20210531 $a 20210602
    - 982 2 2 $z 20210531
  - Retain existing 982; discard the first incoming 982 (since it duplicates an existing one); retain the second incoming 982

- **Example 4**
  - Existing:
    - 982 $a 20210531 $a 20210602
    - 982 2 2 $z 20210531
  - Incoming: no 982s
  - Retain existing 982s
MARC field protection scenarios

From Olamide Kolawole

Scenario 1: MARC Field Entry Is Protected

- Given 5 qualifiers: [Field Number], [Indicator 1], [Indicator 2], [Subfield], [Data]
- And each qualifier can have a wildcard(*) as its value to denote a criteria match of any value
- # definition of each qualifier should be added here
- When a MARC Field entry components matches all 5 qualifiers
  - Then the MARC Field is protected

Scenario 2: MARC Update When an entry already exists exactly, NON-REPEATING

- Given a MARC field entry that exists and is protected and is non-repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When any incoming MARC field entry is exactly the same as the protected existing entry
- Then the incoming MARC field will be discarded

Scenario 3: MARC Update When an entry already exists exactly, REPEATING

- Given a MARC field entry that exists and is protected and is repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When any incoming MARC field entry is exactly the same as the protected existing entry
- Then the incoming MARC field will be discarded

Scenario 4: MARC Update when no entries are the same, NON-REPEATING

- Given a MARC field entry that exists and is protected and is non-repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When no MARC entry in the collection is the same as the existing MARC field entry
- Then the end state will include the existing protected MARC field entry and the collection of incoming MARC field entries

Scenario 5: MARC Update when no entries are the same, REPEATING

- Given a MARC field entry that exists and is protected and is repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When no MARC entry in the collection is the same as the existing MARC field entry
- Then the end state will include the existing protected MARC field entry and the collection of incoming MARC field entries

Scenario 6: MARC Update when the one incoming is the same, NON-REPEATING

- Given a MARC field entry that exists and is protected and is non-repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When at least one field entry in the collection has the same field number as the existing MARC field entry but the whole field entry is not the same
  - Then the end state will include the existing protected MARC field entry and field entries in the collection that do not have the same field number as the existing protected MARC field

Scenario 7: MARC Update when at least one incoming is the same, REPEATING

- Given a MARC field entry that exists and is protected and is repeating
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
- When at least one field entry in the collection has the same field number as the existing MARC field entry but the whole field entry is not the same
  - Then the end state will include the existing protected MARC field entry and the collection of the incoming MARC field entries

Scenario 8: MARC Update when there are no existing records

- Given no MARC field entry exists
- And a collection of incoming MARC field entries that intend to overwrite the existing MARC field entry
Then the end state will include the collection of the incoming MARC field entries

**Overriding Field Protections**