The State University of New York

New MD Editor Series: Normalization Rules (Writing Basic Rules)

Maggie McGee
April 13, 2021

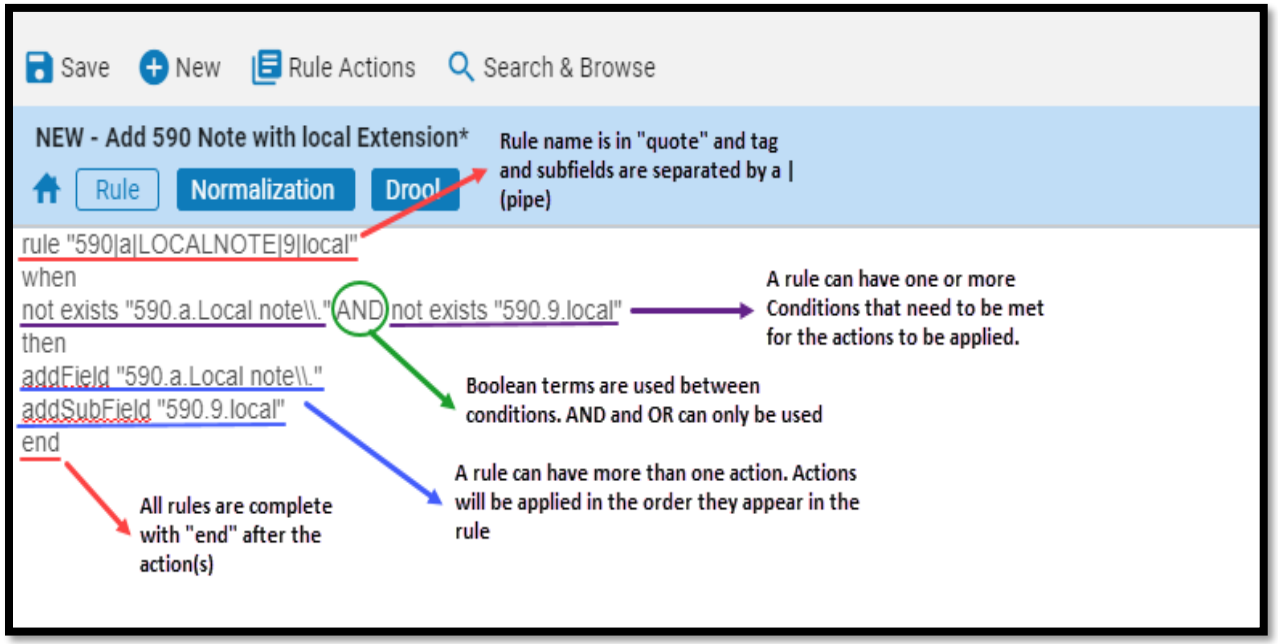
Writing Basics Normalization Rules

This session will cover:

1. Writing basic normalization rules for bib and holdings records:
    1. One condition and one action
    2. One condition and two actions with add a local extension
    3. Two conditions using Boolean terms and actions
    4. Using an “if” statement
2. Testing the Rule in the MD Editor

Review of Normalization Rule Syntax

- Normalization (norm) rules are written using drools logic
- Normalization rules contain:
  - A rule or multiple rules
  - At least one condition
    - Conditions are applied to record elements in a MARC record such as fields, indicators, subfields, field/subfield content
    - Conditions can be defined at the entire rule level (when), or at a specific action level (if)
  - An action or actions
    - Actions tell the rule to do to a MARC21 bib or holdings record



Review of Normalization Rule Syntax

Norm Rule Format & Order	Notes
• **rule** (followed by name of rule) • when • **Condition** or **Conditions** separated by Booelan terms • then • **Action(s)** • end	• All rules must start with rule in the first line • Rule name must be in “quotes” • The rule name cannot use delimiters ($$), instead use a pipe (|) • “Quotes” go around the MARC record information in the condition and action • When, then, and end must be on their own lines • Limit of 200 Boolean terms can be used in a rule • Actions are performed in the order they appear in the rule • Each actions must be on a separate line in the rule • Priorities can be applied when a rule contains multiple rules that apply actions to the same MARC fields

Normalization Rules Syntax

How to construct a rule:

1. The first line begins with **rule** and is followed by the “name of the rule” in quotes
   - Examples;
     - **rule** “remove 597”
     - **rule** “remove 597|a|Estate of Donor Name”
2. **when** comes after the line with rule “rule name”
   - Example;
     - rule “remove 597|a|Estate of Donor Name”
       when
3. **Condition(s) and element(s)** are in the line after **then**
   - Examples;
     - rule “remove 597|a|Estate of Donor Name”
       when
       exists “597”
     - rule “remove 597|a|Estate of Donor Name”
       when
       exists “597.a.Estate of Donor Name*” AND exists “597.d.2016\\.”

Normalization Rules Syntax

How to construct a rule with one rule and one condition continued:

4. **then** comes after the line with the condition(s)
   - Example;
     - rule “remove 597|a|Estate of Donor Name”
       when
       exists “597.a.Estate of Donor Name*” AND exists “597.d.2016\\.”
       then
5. The **action(s) and element(s)** is in the line after then
   - Example;
     - rule “remove 597|a|Estate of Donor Name”
       when
       exists “597.a.Estate of Donor Name*” AND exists “597.d.2016\\.”
       then
       removeField “597”

Normalization Rules Syntax

How to construct a rule with one rule and one condition continued:

6. **end** comes after the action(s) and element(s) line(s)
    - Example;
        - rule “remove 597|a|Estate of Donor Name”
          when
          exists “597.a.Estate of Donor Name*” AND exists “597.d.2016\\.”
          then
          removeField “597”
          end

Placement of Normalization Rules

- The placement of new rules is very important when creating normalization (norm) rules because it will impact what processes and functionality can use the norm rules in Alma:
  - The "Placement of new rules" should be configured to local when creating or duplicating norm rules
  - A Network (network icon) or a local (house icon) norm rule can be used on a bib record in the MD Editor, but a Network created norm rule cannot be used in a norm process
  - Local rules can only be used on IZ only records when using "Enhance the record" in the MD Editor
  - Network rules can only be used on NZ linked bib records when using "Enhance the record" in the MD Editor

Placement of Normalization Rules

To change the “Placement of new rules”:

1. Open the MD Editor
2. Click on the Records or Template tab
   - Currently Placement Options is not under the "New" menu when the Rules tab is selected
3. Click ***New>Placement Options***
4. Click the radio button next to **Local** under "Placement of new rules“
5. ***Click Save***

CONDITIONS

SUNY The State University of New York

Normalization Rule Conditions

Condition	Applied to Record Elements
containsScript	Fixed list of languages
exists	Data fields: • MARC fields/subfield • Not applied to fixed fields
existsControl	Control fields (fixed fields): • LDR • 00X fields
not exists	Data fields: • MARC fields/subfield • Not applied to fixed fields
not existsControl	Control fields (fixed fields): • LDR • 00X fields
existsMoreThanOnce	Data fields: • MARC fields/subfield • Not applied to fixed fields
TRUE	True is a condition used at the record level • Is an absolute condition • "If" statements in the action are used to limit when the action is applied

ACTIONS

Control Field Actions – Only Apply to IZ Only Records

Rule Action	Definition	Example
changeControlField	• Changes the MARC tag for control fields only • Does not change values of the control fields • **NOT USED IN SUNY for NZ linked records**	changeControlField "<Fixed Field MARC Tag>" to "<New Fixed Field Tag>“ changeControlField "007" to "008"
addControlField	• Adds a fixed field to a MARC record • Requires the Control tag, indicators, and value to be part of the action • **NOT USED IN SUNY for NZ linked records**	addControlField "<Fixed Field Tag>.<Value>" addControlField "007.vd cvaizm • The space in the "quotes" of the value represents a blank
removeControlField	• Removes all occurrences of the fix field • Alma will automatically replace 008 if it is removed, suggest to immediately adding an 008 in the rule after removing it • **NOT USED IN SUNY for NZ linked records**	removeControlField "<Fixed Field Tag>" removeControlField "003" removeControlField "008" addControlField "008.180620s2019 ncua b s001 0beng c" • The spaces represents a blank
replaceControlContents	• Replaces fixed field values • Replaces only fixed field values that match the action • **NOT USED IN SUNY for NZ linked records**	replaceControlContents "<Fixed Field Tag>.{<Position,Length>}.<Value>" with "<New Value>" replaceControlContents "LDR.{17,1}.l" to " " • The space represents a blank

Data Field Actions

- Use the data field actions in:
  - IZ Only records for any MARC field
    - **Do not** make changes to non-local MARC if the IZ only record is going to be linked to the NZ
  - NZ linked bib records for SUNY defined local MARC fields
    - SUNY defined local fields are available here: https://slcny.libguides.com/ld.php?content_id=49705411

- The following actions are available for use in Alma as of March April 2021:
  1. addCreatingAgency
  2. addModifyingAgency
  3. replaceModifyingAgency

Data Field Actions – Add Fields and Subfields

Record Action to Add Fields and Subfields	Definition	Example
addField	• Adds the MARC field • Requires indicators or subfields and a value to be a part of the action in order to add the MARC field	addField "<MARC Tag>.{<Indicator 1,Indicator 2>}.<Subfield>.<Value>" or addField "<Marc Tag>.<Subfield>.<Value>“ addField "597.{1, }.c.Gift addField “597.c.Gift
addSubField	• Adds a subfield and value to an existing MARC field • Requires the MARC tag, subfield, and value to be part of the action	addSubField "<MARC Tag>.{<Indicator 1,Indicator 2>}.<Subfield>.<Value>" or addSubField "<MARC Tag>.<Subfield>.<Value>" addSubField "597.{*,*}.9.local" addSubField "597.9.local"
addSystemNumber	• Adds the system number from one field to another field • Requires	addSystemNumber "<MARC Tag where the System Number will be added><Subfield to place the System Number>" from "<MARC Tag where the System Number is being copied from>" prefixed by "<MARC Tag with the Prefix> addSystemNumber "035.a" from "001" prefixed by "003"

Data Field Actions – Add Fields and SubFields

Record Action to Copy Fields	Definition	Example
prefix	• Adds a prefix to the value of a subfield	prefix "<MARC Tag>.<Subfield>" with "<Value>" prefix "035.b" with "(OCoLC)"
prefixSubField	• Adds the prefix of the value in one subfield with the prefix value of another subfield	prefixSubField "<MARC Tag>.<Subfield>" with "<Source MARC Tag>.<Source Subfield>" prefixSubField "035.b" with "035.a"
suffix	• Add a suffix to the end of a value	suffix "<MARC Tag>.<Subfield>" with "<Value>" suffix "953.a" with "-NEW"
suffixSubField	• Adds the suffix of the value in one subfield with the suffix value of another subfield	suffixSubField "<MARC Tag>.<Subfield>" with "<Source MARC Tag>.<Source Subfield>" suffixSubField "935.a" with "035.a"

Data Field Actions – Copy Fields

Record Action to Copy Fields	Definition	Example
copyField	• Copies the field to another field • Creates a separate field if it already exists • The original field, subfield, and value are retained • Adding subfields will only copy the defined subfield to the new subfield • Requires removeField to remove the original field	copyField "<MARC Tag>" to "<MARC Tag>" or copyField "<MARC Tag>.<Subfield>" to "<MARC Tag>.<Subfield>" copyField "597" to "590" copyField "597.c" to "590.a"

Data Field Actions – Replace Fields and Subfields

Record Action to Replace Field & Subfields	Definition	Example
changeField	• Changes the MARC tag for data fields • Does not change indicators, subfields, or values	changeField "<MARC Tag>" to "<New MARC Tag>" changeField "500" to "505"
changeSubField	• Changes the subfield only • Will change all duplicate subfields to the new subfield • Does not change the value in the subfields	changeSubField "<MARC Tag>.<Subfield>" to "<New Subfield>" changeSubField "597.a" to "b"
changeSubFieldOnlyFirst	• Changes the first subfield only • Will not change duplicate subfields • Does not change the value(s) of the first subfield	changeSubFieldOnlyFirst "<MARC Tag>.<Subfield>" to "<New Subfield>" changeSubFieldOnlyFirst "597.a" to "b"
changeSubFieldExceptFirst	• Changes all subfields except the first subfield with duplicate values • Does not change the values in all subfield except the first	changeSubFieldExceptFirst "<MARC Tag>.<Subfield> to "<New SubField>" changeSubFieldExceptFirst "597.a" to "b"

Data Field Actions – Replace Fields and Subfields

Record Action to Replace Field & Subfields	Definition	Example
changeFirstIndicator	• Changes the value of the first indicator only	changeFirstIndicator "<MARC Tag>" to "<Value>" changeFirstIndicator "597" to "0"
changeSecondIndicator	• Changes the value of the second indicator only	changeSecondIndicator "<MARC Tag>" to "<Value>" changeSecondIndicator "597 to " " • The space between the "quotes" represents a blank
combineFields	• Combines duplicate MARC fields • Excludes copying subfields after the first occurrence of the excluded subfield(s) have been copied	combineFields "<MARC Tag>" excluding "<Subfield>" or combineFields "<MARC Tag>" excluding "<Subfield,Subfield>“ combineFields "597" excluding "c"

Data Field Actions – Remove Fields and Subfields

Record Action to Remove Fields and Subfields	Definition	Example
removeField	• Removes all occurrences of the field • “if” statements in the action can restrict removing the field only if it meets the criteria in the “if” statement	removeFied "<MARC Tag>" removeField "986" removeField “986” if(exists “986.a.NEW*”)
removeSubField	• Removes all occurrences of the subfield from the indicated MARC field	removeSubField "<MARC Tag>.<Subfield>" removeSubfield "590.b"

Data Field Actions – Replace Values in Fields and Subfields

Record Action to Replace Values in Fields and Subfields	Definition	Example
replaceContents	• Replaces all occurrences of the matching values with the new value	replaceContents "<MARC TAG>.<Subfield>.<Value>" with "<New Value>" replaceContents "597.c.Gift" with "Purchase"
replaceContentsOnlyFirst	• Replaces the contents of the first matching value only	replaceContentsOnlyFirst "<MARC Tag>.<Subfield>.<Value>" with "<New Value>" replaceContentsOnlyFirst "592.a.Thesis" with "Dissertation"
replaceContentsExceptFirst	• Replaces matching value occurrences for all but the first matching value	replaceContentsExceptFirst "<MARC Tag>.<Subfield>.<Value>" with "<New Value>“ replaceContentsExceptFirst “590.a.“committed to retain” with "condition reviewed"
replaceSubFieldContents	• Replaces all occurrences of the subfield with the contents of the defined subfield	replaceSubFieldContents "<MARC Tag>.<Subfield>" with "<MARC Tag>.<Subfield>" replaceSubfieldContents "500.a" with "590.b"

Data Field Actions – Other Actions

Record Action	Definition	Example
splitSubField	• Splits the subfield to the new MARC Tag/subfield based on the action • The original field remains intact • The whole subfield will be copied if the “to” subfield does not exist • The “to” part of the action is not required • The “to” statement will creates a field/subfield for each occurrence of the defined punctuation • Indicators or optional • addSeq will place the order of the of the split subfields • addSeq is optional	splitSubField “<MARC Tag>.<{Indicator 1,Indicator 2}>.<Subfield>.<Punctuation>” to “<MARC Tag>>.<{Indicator 1,Indicator 2}>.<Subfield>” addSeq “<Subfield>” splitSubField “520.a.,” to “590.a” splitSubField “520.a.,” to “590.a” addSeq “8”
correctDuplicateFields	• Removes duplicate fields • First occurrence is retained, and subsequent duplicate fields are deleted • Commas are used between the MARC tags in the action	correctDuplicateFields "<MARC Tag>,<MARC Tag>" correctDuplicateFields "650,655"
correctDuplicateSubfields	• Keeps the first occurrence of the duplicate subfield • Deletes the subsequent occurrences of the duplicate subfield	correctDuplicateSubfields "<MARC Tag>" "<Subfields>" correctDuplicateSubfields "690" "z" correctDuplicateSubfields "690" "x,y,z"

Data Field Actions – Other Actions

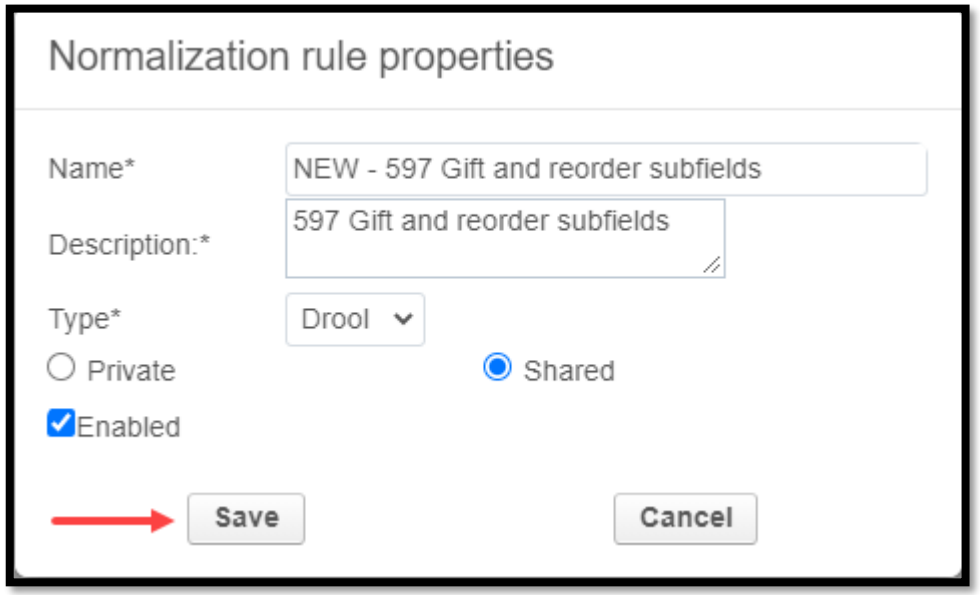
Record Action	Definition	Example
moveSubfieldsToTheEndOfField	• Moves the subfield to the end of the field • Removes other occurrences of the same subfield • More than one subfield moved to the end, they will appear in the order of the action • Subfields do not contain commas between them • "If" statements are not supported with this action	moveSubfieldsToTheEndOfField "<MARC Tag>" "<Subfield or Subfields>" moveSubfieldsToTheEndOfField "690" "z" moveSubfieldsToTheEndOfField "690" "z,2"

WRITING AND TESTING A NORMALIZATION RULE WITH ONE CONDITION AND ONE ACTION

Normalization Rules with One Condition and One Action

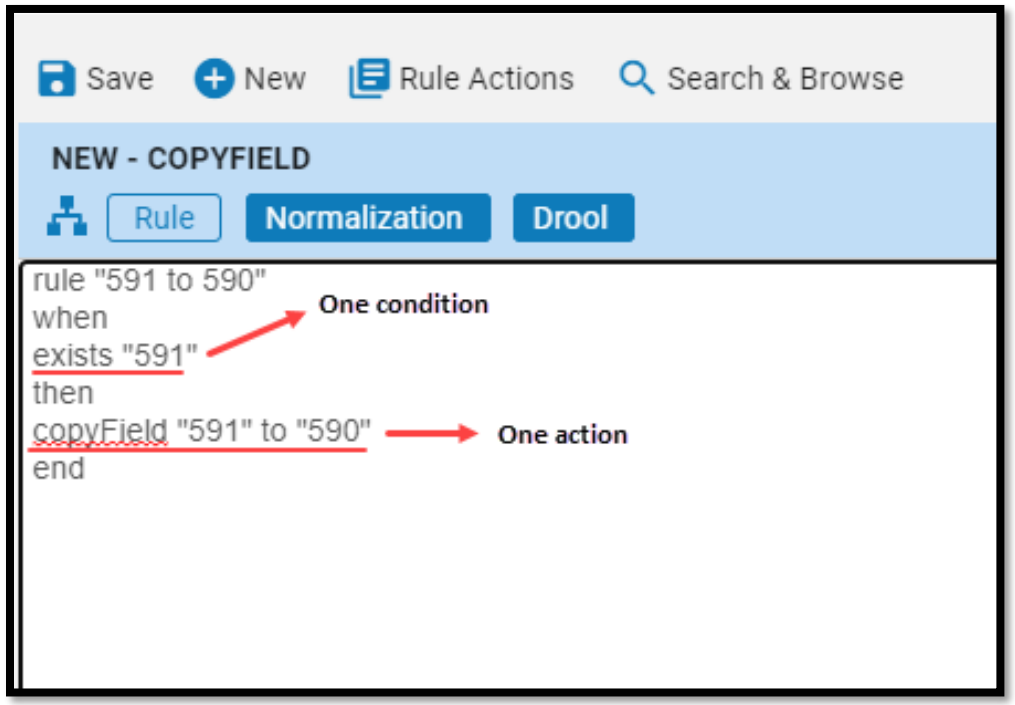
To write a norm rule with one condition and action:

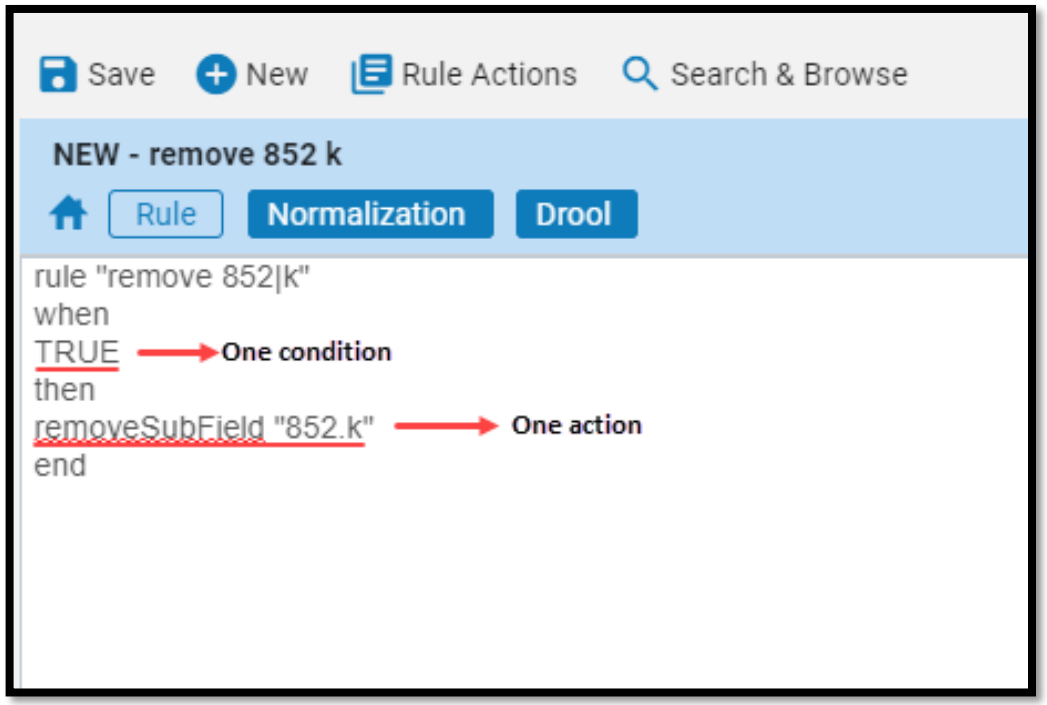
1. Open the MD Editor
2. Check the “Placement of new rules” is set to local
3. Click on the Rules tab
4. Click ***New>Normalization*** from the main menu
5. The Normalization Rules Properties Screen opens
    1. Name: [Name the Rule using the SUNY naming convention]
    2. Description: [Optional]
    3. Type: [Keep the default value Drool]
    4. Shared
    5. Enabled
    6. Click ***Save***
6. Begin writing the rule
7. Click ***Save***



Normalization Rules with One Condition and One Action

- Examples of a rule with one condition and one action:

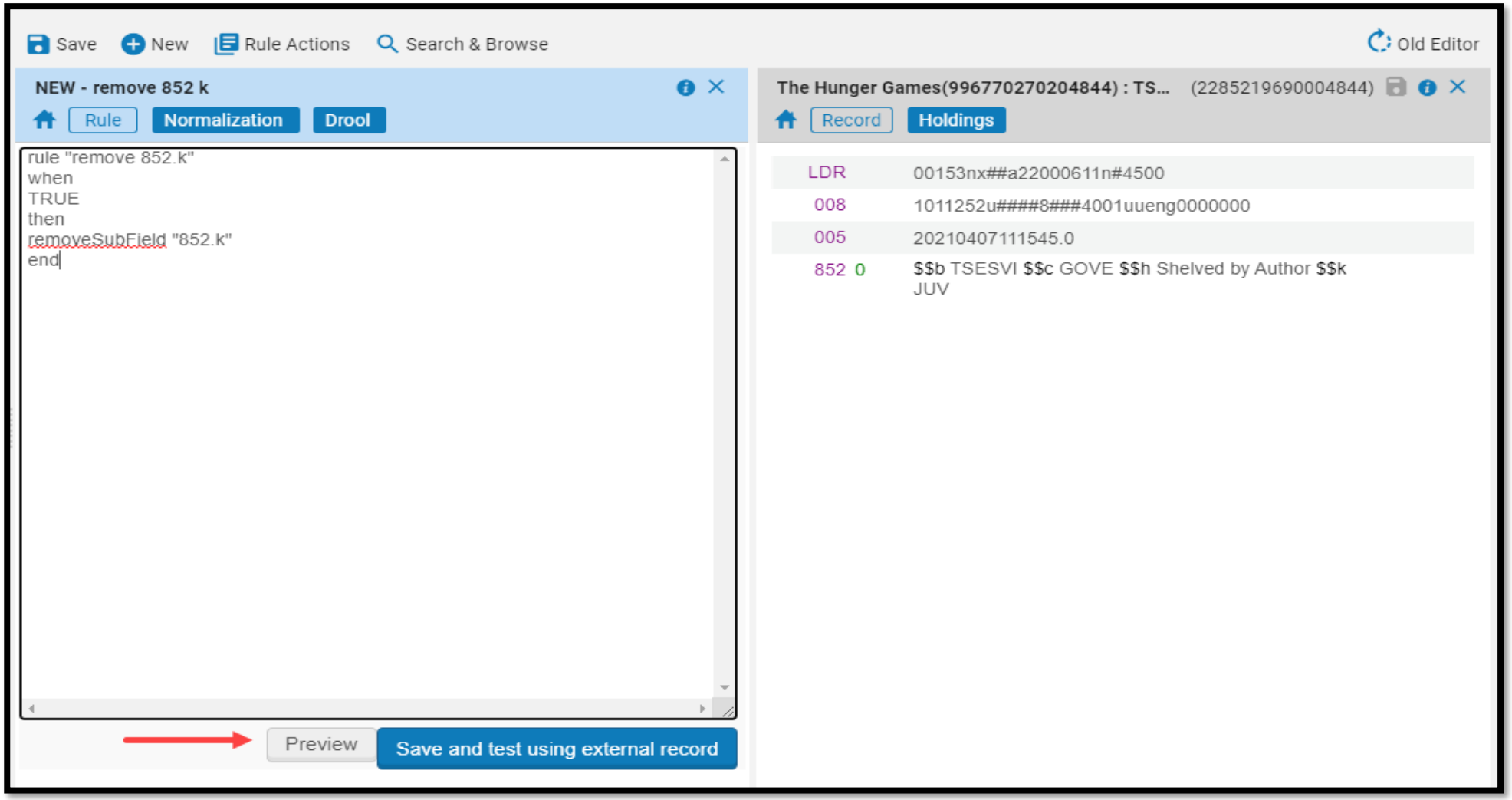




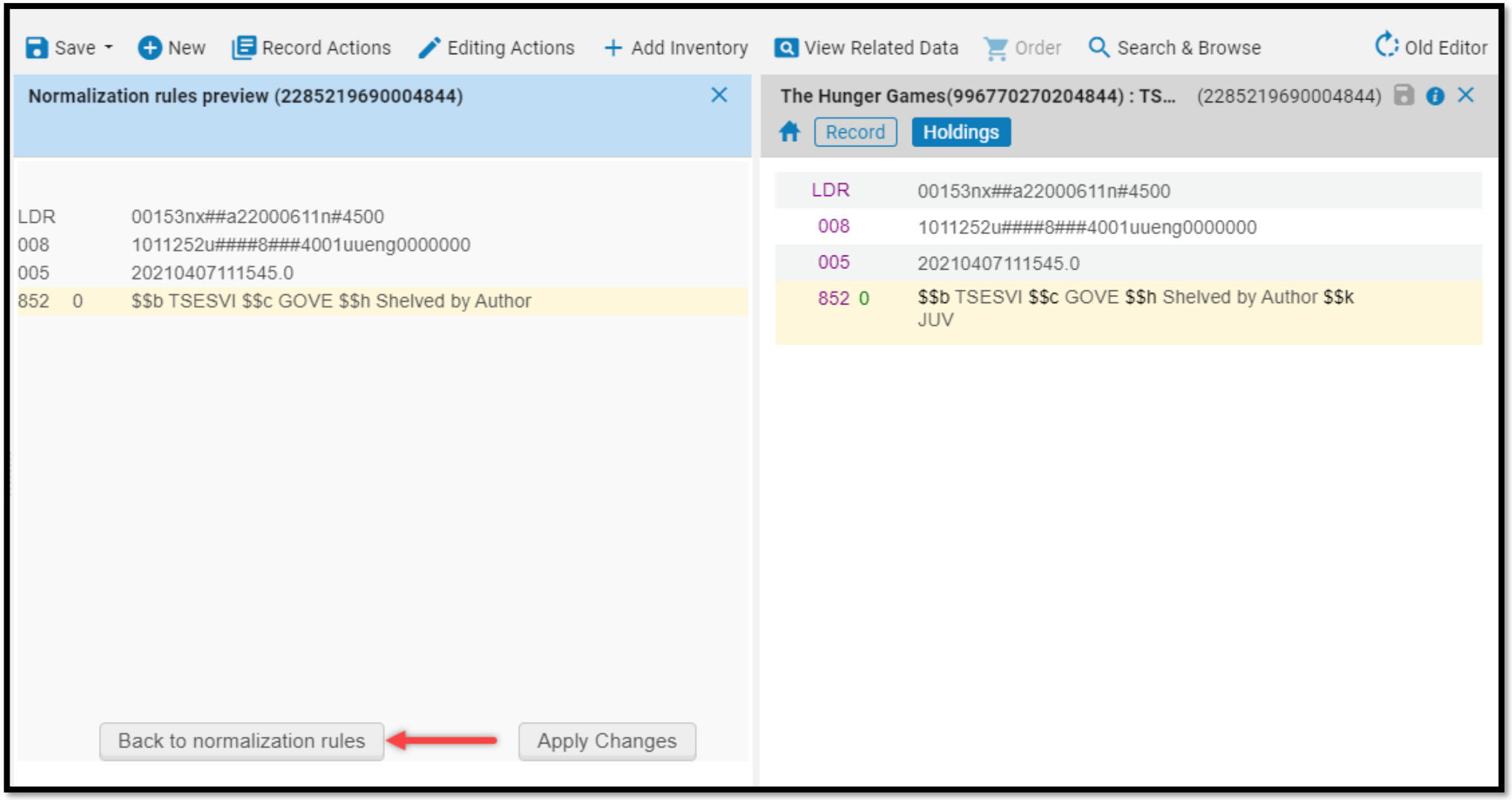
Testing a Norm Rule in the MD Editor

- With the norm rule open in the MD Editor:
  1. Click on the Editor split mode icon (F6)
  2. Click the **Records** tab
  3. Click on the **MARC21 Bib** record type
  4. Search for the bib record
  5. **Left-click** to open the bib record on the right-hand side of the main pane
  6. Click on the rule on the left-side of the main pane
  7. Click ***Preview***
     - A preview of the rule changes will display
     - Click Back to Normalization Rule
  8. Edit, preview, and save the rule if the rule did not work as expected in step 9
  9. Click "***X***" to close the indication rule
  10. Release the bib record (***Records Actions>Release Record*** / Alt+Shft+R)

Testing a Norm Rule in the MD Editor



Testing a Norm Rule in the MD Editor



Try Creating and Testing a Norm Rule

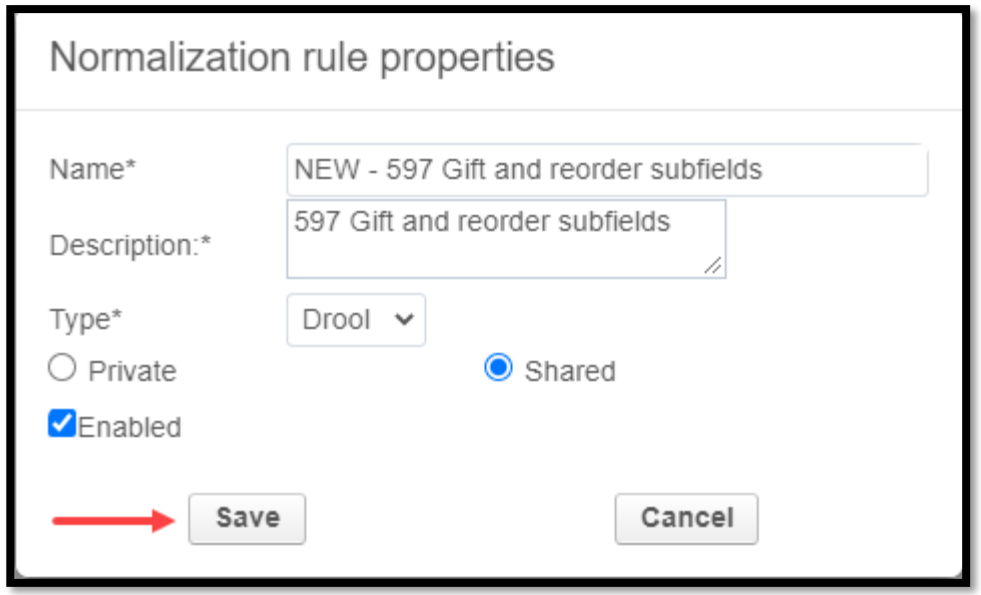
1. Create a holdings norm rule that meets the following criteria:
    1. Change MARC 852 $$c from one location to another location in your institution
    2. The location being changed must be in the condition element
    3. The location it is being changed to will be in the action element using:
        - replaceContents
2. Test the norm rule in the MD Editor on a holdings record that:
    1. Contains the location in 852 $$c being changed
    2. Does not contain the location in 852 $$c being changed

WRITING AND TESTING A NORMALIZATION RULE WITH ONE CONDITION AND TWO ACTIONS

Writing a Normalization Rule with One Condition & Two Actions

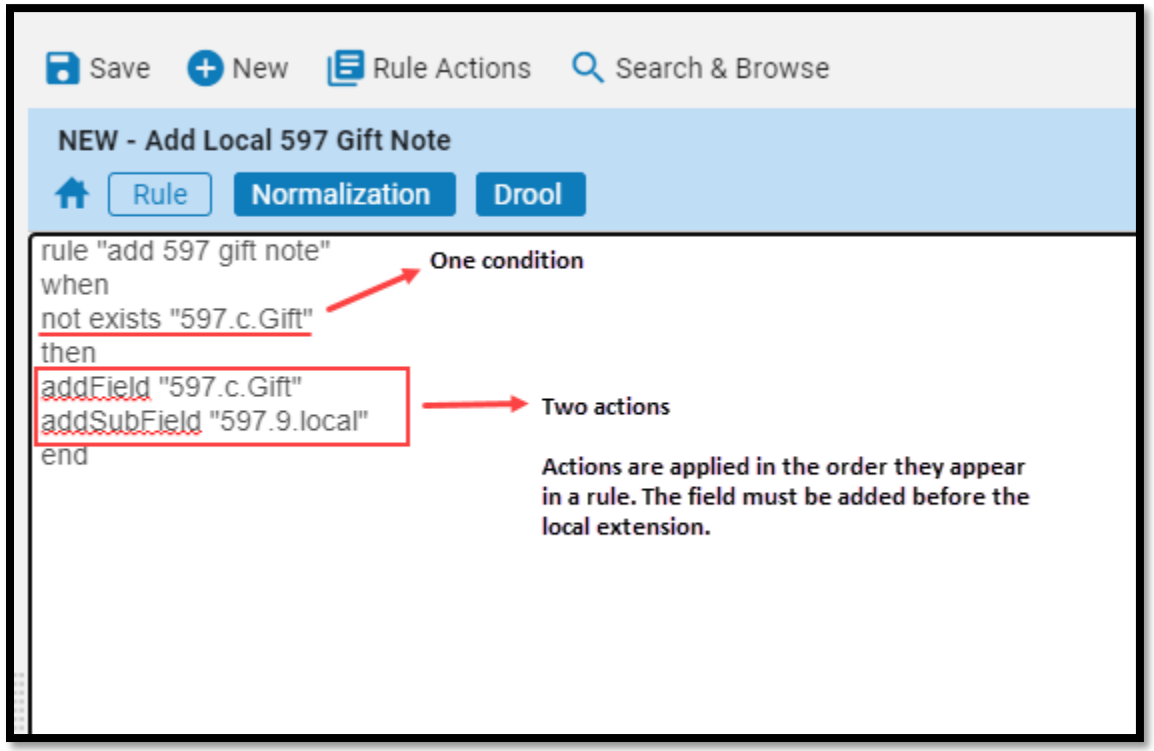
To write a norm rule with one condition and action:

1. Open the MD Editor
2. Check the “Placement of new rules” is set to local
3. Click on the Rules tab
4. Click ***New>Normalization*** from the main menu
5. The Normalization Rules Properties Screen opens
    1. Name: [Name the Rule using the SUNY naming convention]
    2. Description: [Optional]
    3. Type: [Keep the default value Drool]
    4. Shared
    5. Enabled
    6. Click ***Save***
6. Begin writing the rule
7. Click ***Save***



Writing a Normalization Rule with One Condition & Two Actions

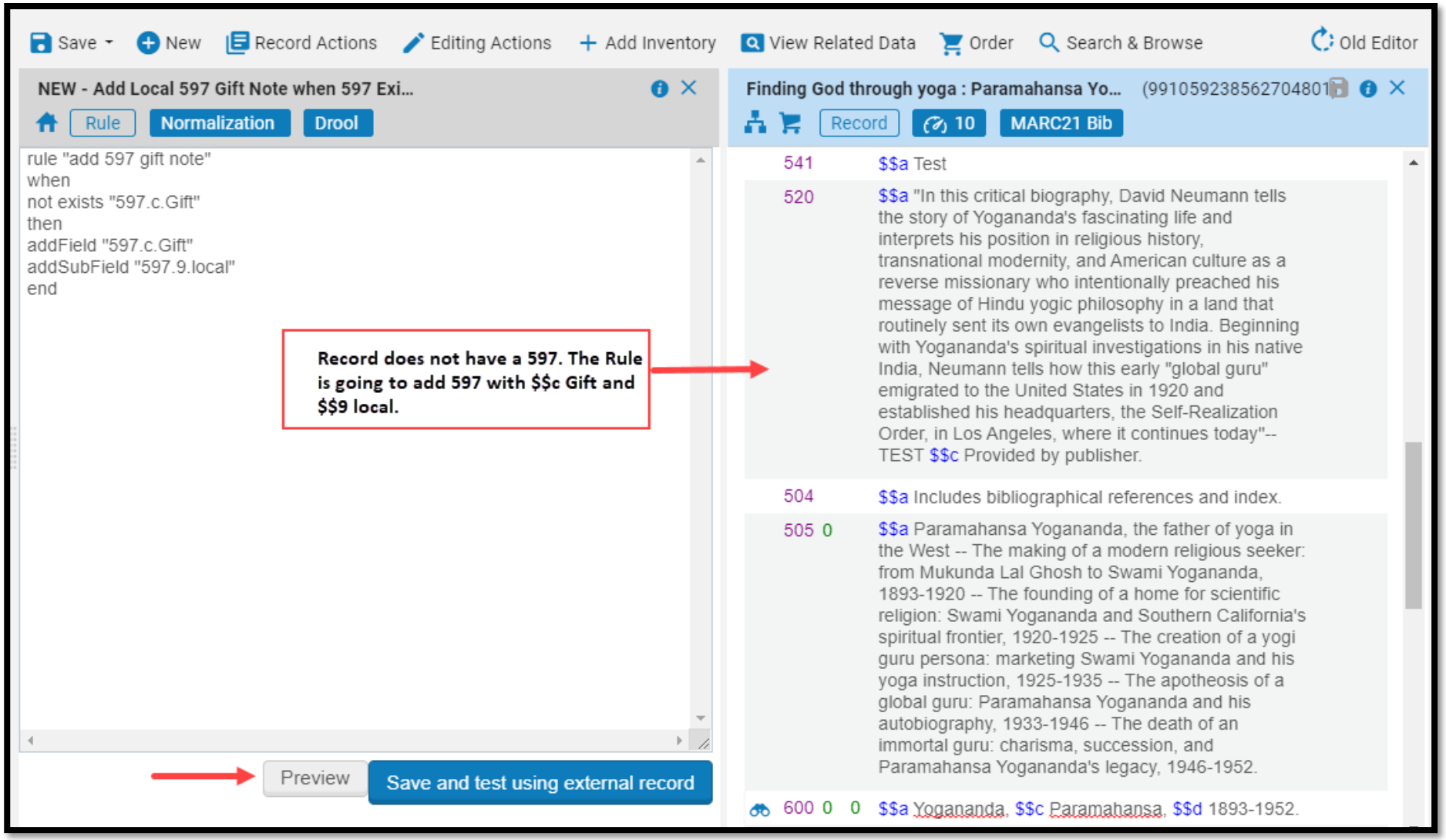
- Example of a rule with one condition and two actions:
  - The example below is how to add a SUNY local MARC field with a local extension
  - Actions are performed in the order they appear:
    - The field must be created first and then $$9 local can be added



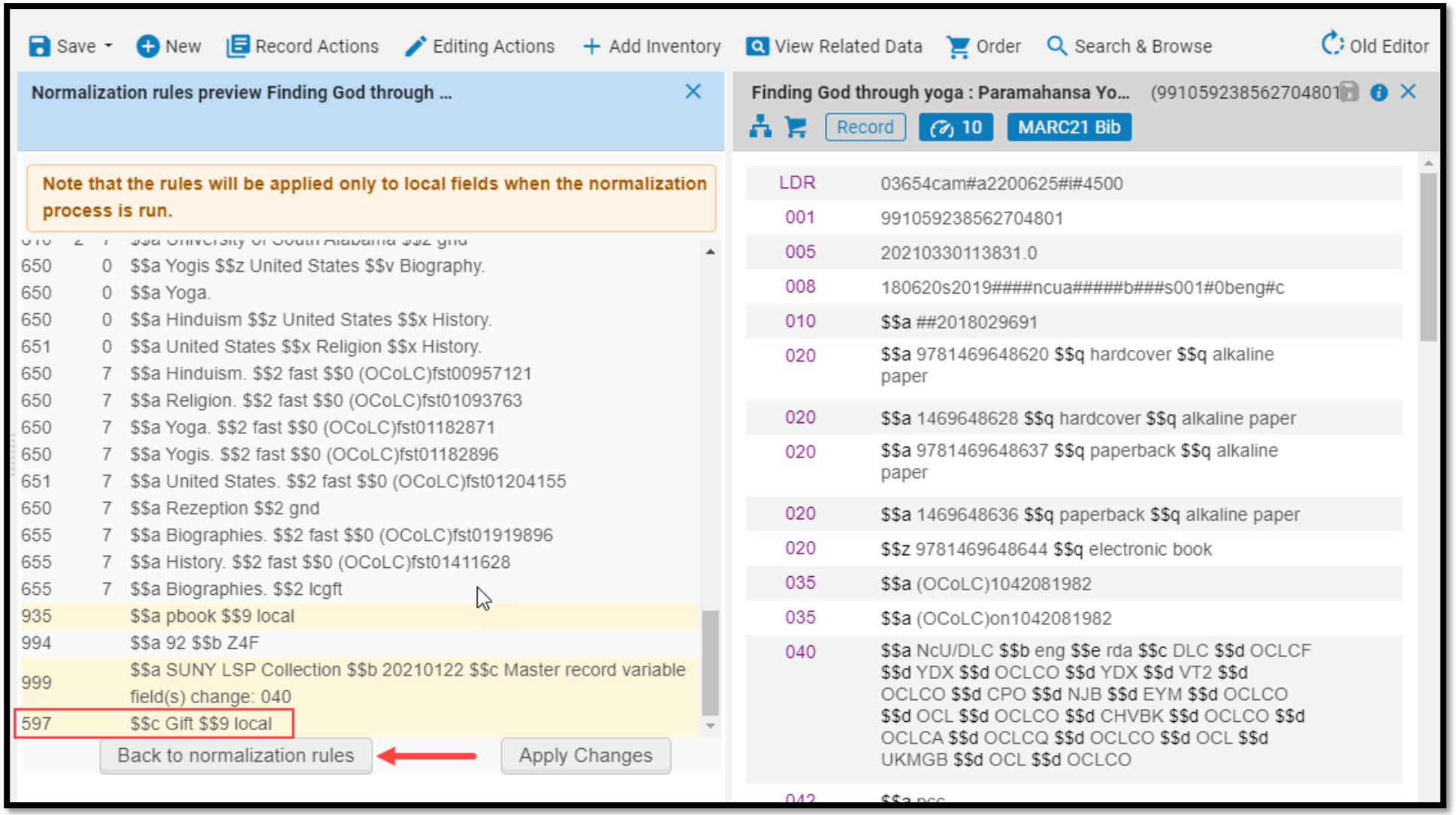
Testing a Norm Rule in the MD Editor

- With the norm rule open in the MD Editor:
  1. Click on the Editor split mode icon (F6)
  2. Click the **Records** tab
  3. Click on the **MARC21 Bib** record type
  4. Search for the bib record
  5. **Left-click** to open the bib record on the right-hand side of the main pane
  6. Click on the rule on the left-side of the main pane
  7. Click ***Preview***
     - A preview of the rule changes will display
     - Click Back to Normalization Rule
  8. Edit, preview, and save the rule if the rule did not work as expected in step 9
  9. Click "***X***" to close the indication rule
  10. Release the bib record (***Records Actions>Release Record*** / Alt+Shft+R)

Testing a Norm Rule in the MD Editor



Testing a Norm Rule in the MD Editor



Try Creating and Testing a Norm Rule

1. Create a bib norm rule that meets the following criteria:
    1. MARC 590 does not already exist
    2. Add a Local MARC field 590 with $$a Normalization Rules Training Session Test
    3. and $$9 local using:
        - addField
        - addSubField
2. Test the norm rule in the MD Editor with a bib record that:
    1. Contains an existing 590
    2. Does not contain an existing 590

WRITING AND TESTING A NORMALIZATION RULE WITH TWO CONDITIONS, BOOLEAN TERMS, AND ACTIONS

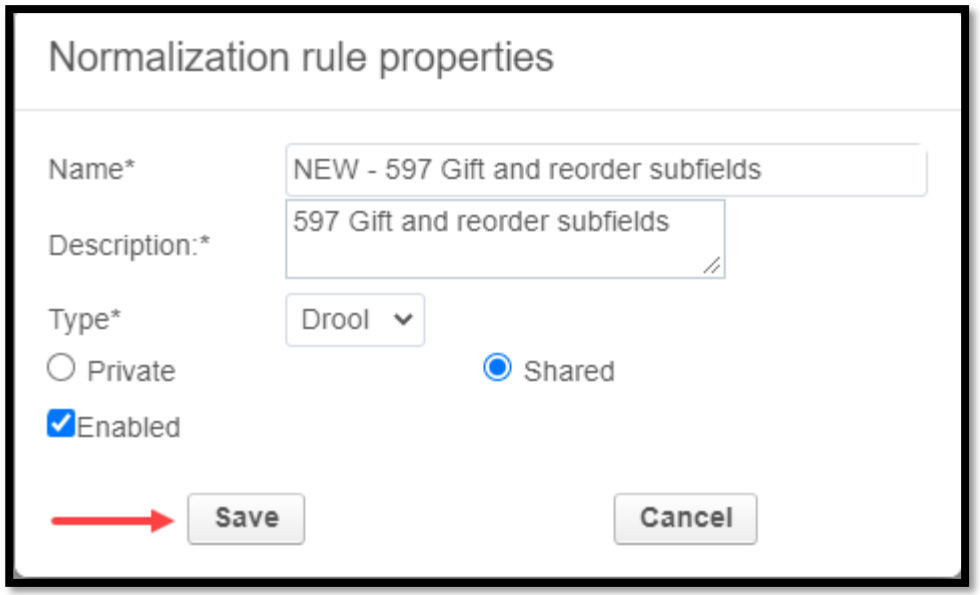
SUNY The State University of New York

Normalization Rules with Two Conditions, Boolean Terms, and Actions

- The following Boolean operators can be used in norm rules:
  - AND
  - OR
- The AND Boolean operator means that all conditions must be met for the action(s) to be applied
  - Example: exists "852.b.NEWMN" AND exists "852.c.STACK“
    - The first and second condition must be met for the action(s) to be applied
    - 852 $$b NEWMN and 852 $$c STACK must exist
- The OR Boolean operator means one of the conditions must be met for the action(s) to be applied
  - Example: exists “852.c.STACK” OR exists “852.c.GOVE”
    - One of the conditions must be met for the action(s) to be applied
    - Either 852 $$c STACK or 852 $$c GOVE exists
- Boolean terms can be combined
  - (Parenthesis) are helpful to group conditions/elements using different Boolean operators
  - Example: (exists "852.c.GOVE" AND not exists "852.k.JUV") OR (exists "852.c.JUV" AND not exists "852.k.JUV")
    - (must contain 852 $$c GOVE and not contain 852 $$k JUV) or (must contain 852 $$c JUV, but not contain 852 $$k JUV)

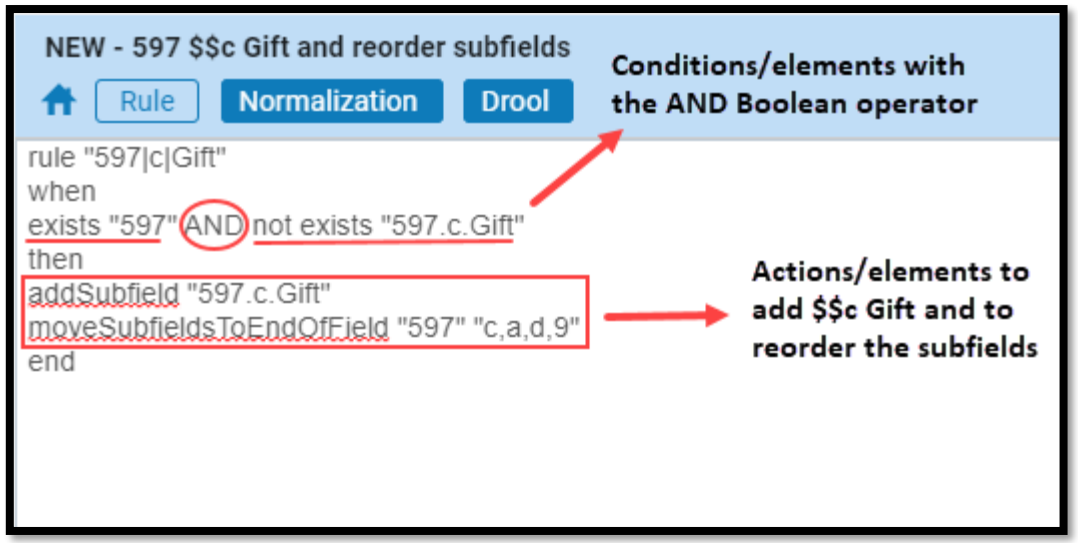
Normalization Rules with Two Conditions, Boolean Terms, and Actions

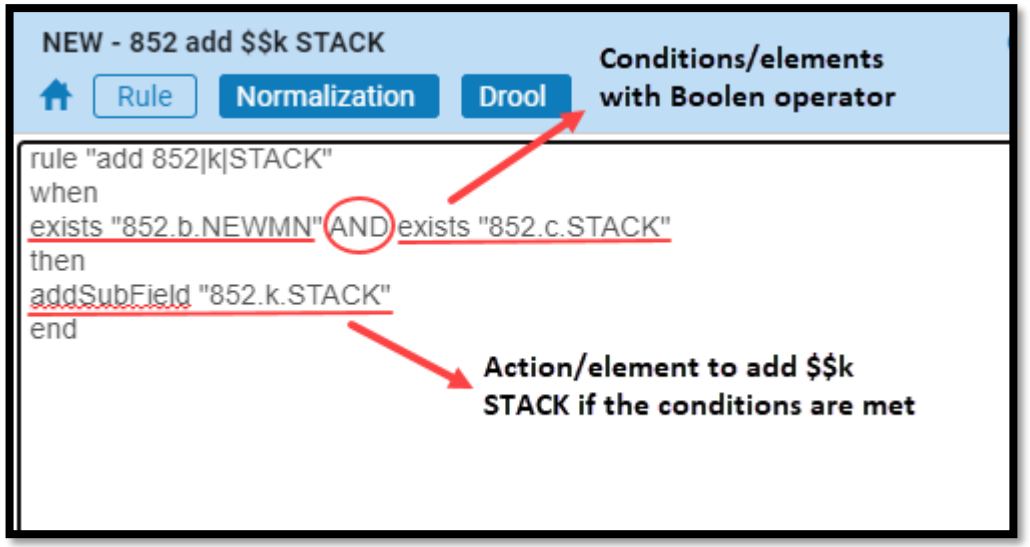
- To write a norm rule with one condition and action:
  1. Open the MD Editor
  2. Check the “Placement of new rules” is set to local
  3. Click on the Rules tab
  4. Click ***New>Normalization*** from the main menu
  5. The Normalization Rules Properties Screen opens
     1. Name: [Name the Rule using the SUNY naming convention]
     2. Description: [Optional]
     3. Type: [Keep the default value Drool]
     4. Shared
     5. Enabled
     6. Click ***Save***
  6. Begin writing the rule
  7. Click ***Save***



Normalization Rules with Two Conditions, Boolean Terms, and One Action

- Examples of norm rules with two conditions, Boolean terms, and actions:

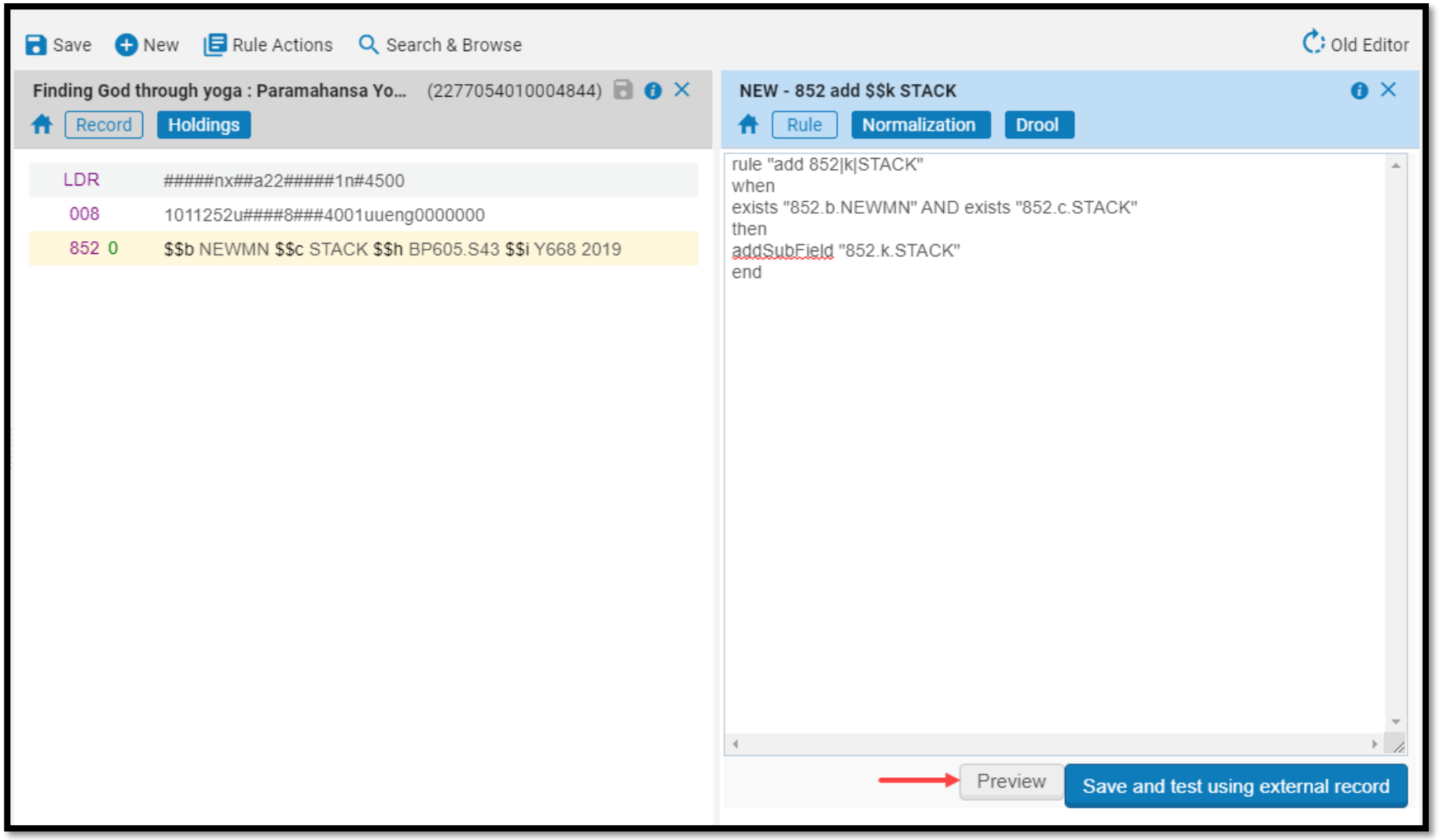




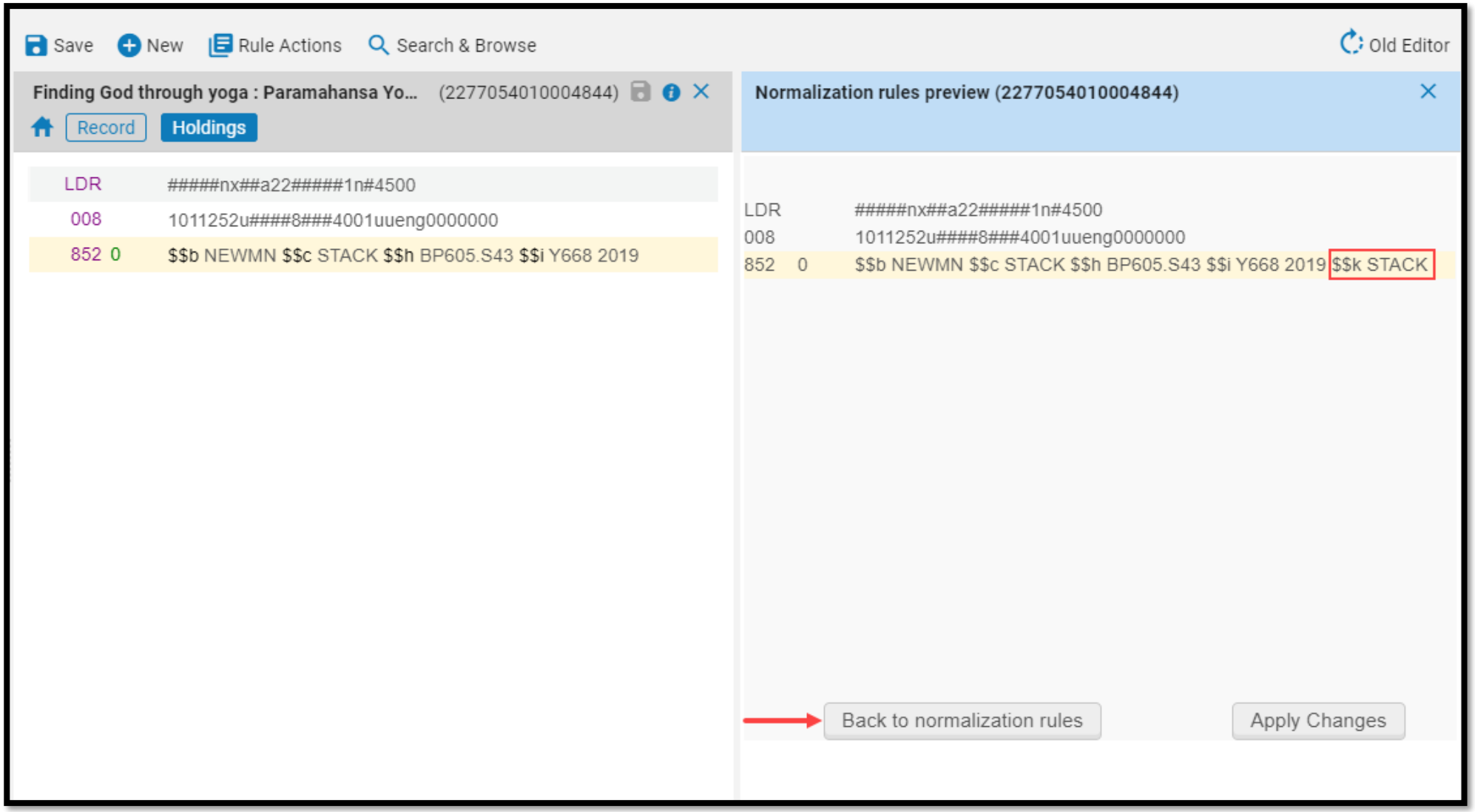
Testing a Norm Rule in the MD Editor

- With the norm rule open in the MD Editor:
  1. Click on the Editor split mode icon (F6)
  2. Click the **Records** tab
  3. Click on the **MARC21 Bib** record type
  4. Search for the bib record
  5. **Left-click** to open the bib record on the right-hand side of the main pane
  6. Click on the rule on the left-side of the main pane
  7. Click ***Preview***
     - A preview of the rule changes will display
     - Click Back to Normalization Rule
  8. Edit, preview, and save the rule if the rule did not work as expected in step 9
  9. Click "***X***" to close the indication rule
  10. Release the bib record (***Records Actions>Release Record*** / Alt+Shft+R)

Testing a Norm Rule in the MD Editor



Testing a Norm Rule in the MD Editor



Try Creating and Testing a Norm Rule

1. Create a holdings norm rule that meets the following criteria:
    1. Add $$k to MARC 852
        1. Use addSubField
    2. Choose a $$k value used at your institution or make one up if $$k is not used
    3. Choose the conditions that must be met to add $$k
    4. Use one Boolean operator in the conditions/element line of the rule
2. Test the norm rule in the MD Editor with a holding record that:
    1. Meets the conditions
    2. Does not meet the conditions

WRITING AND TESTING A NORMALIZATION RULE WITH AN "IF" STATEMENT

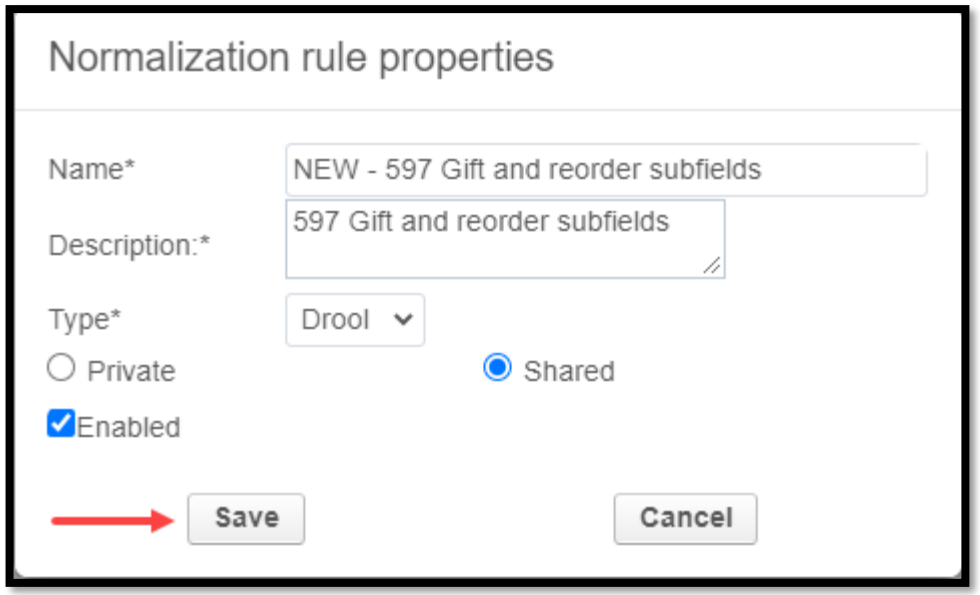
SUNY The State University of New York

Normalization Rule with an “if” Statement

- An "if" statements is a condition that is combined with an action:
  - The action will be applied to the field when the condition in the "if" statement is met
  - The action will not be applied to the field when the condition in the "if" statement conditions is not met
- "if" statements can be used with a condition containing elements or with TRUE
- "if" statements can be used with all actions except, **moveSubfieldsToEndOfField**
- "if" statements cannot have more than one condition and element
- “if” statements can be used to make the rule more granular so changes are only made “if” the condition is met

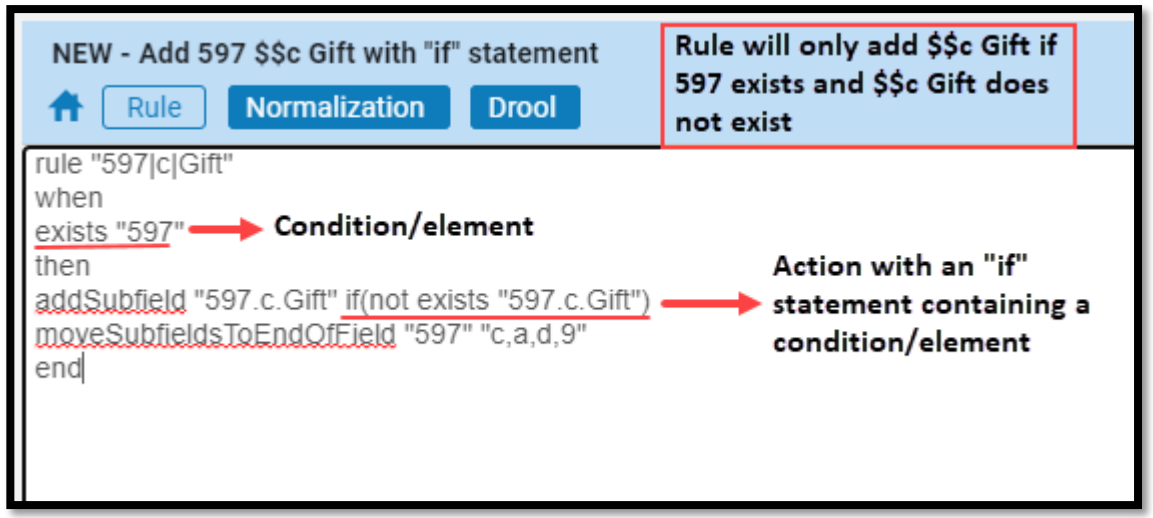
Normalization Rule with an “if” Statement

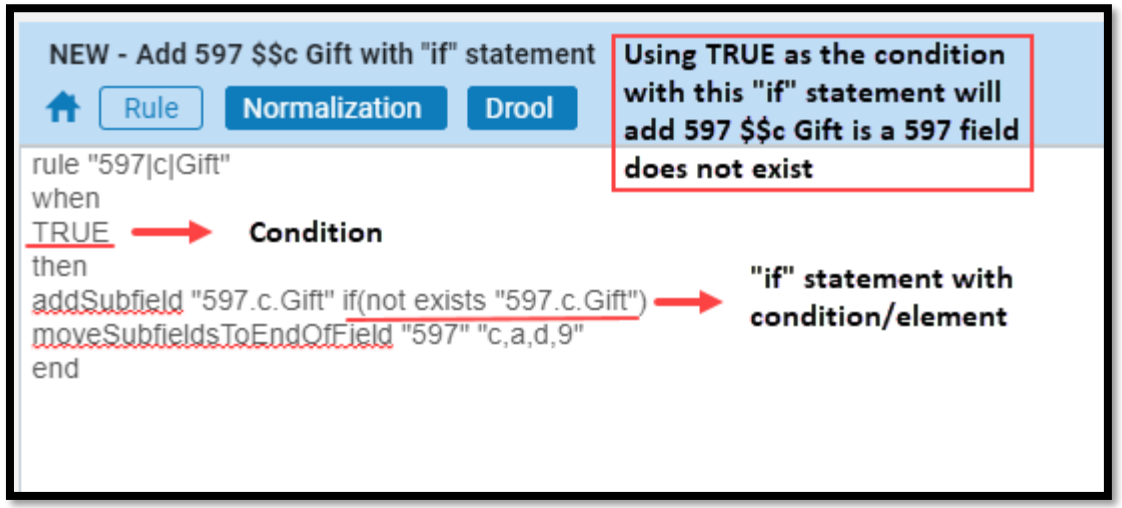
- To write a norm rule with one condition, action, and “if” statement:
  1. Open the MD Editor
  2. Check the “Placement of new rules” is set to local
  3. Click on the Rules tab
  4. Click ***New>Normalization*** from the main menu
  5. The Normalization Rules Properties Screen opens
     1. Name: [Name the Rule using the SUNY naming convention]
     2. Description: [Optional]
     3. Type: [Keep the default value Drool]
     4. Shared
     5. Enabled
     6. Click ***Save***
  6. Begin writing the rule
  7. Click ***Save***



Normalization Rule with an “if” Statement

- Examples of norm rules with an “if” statement :

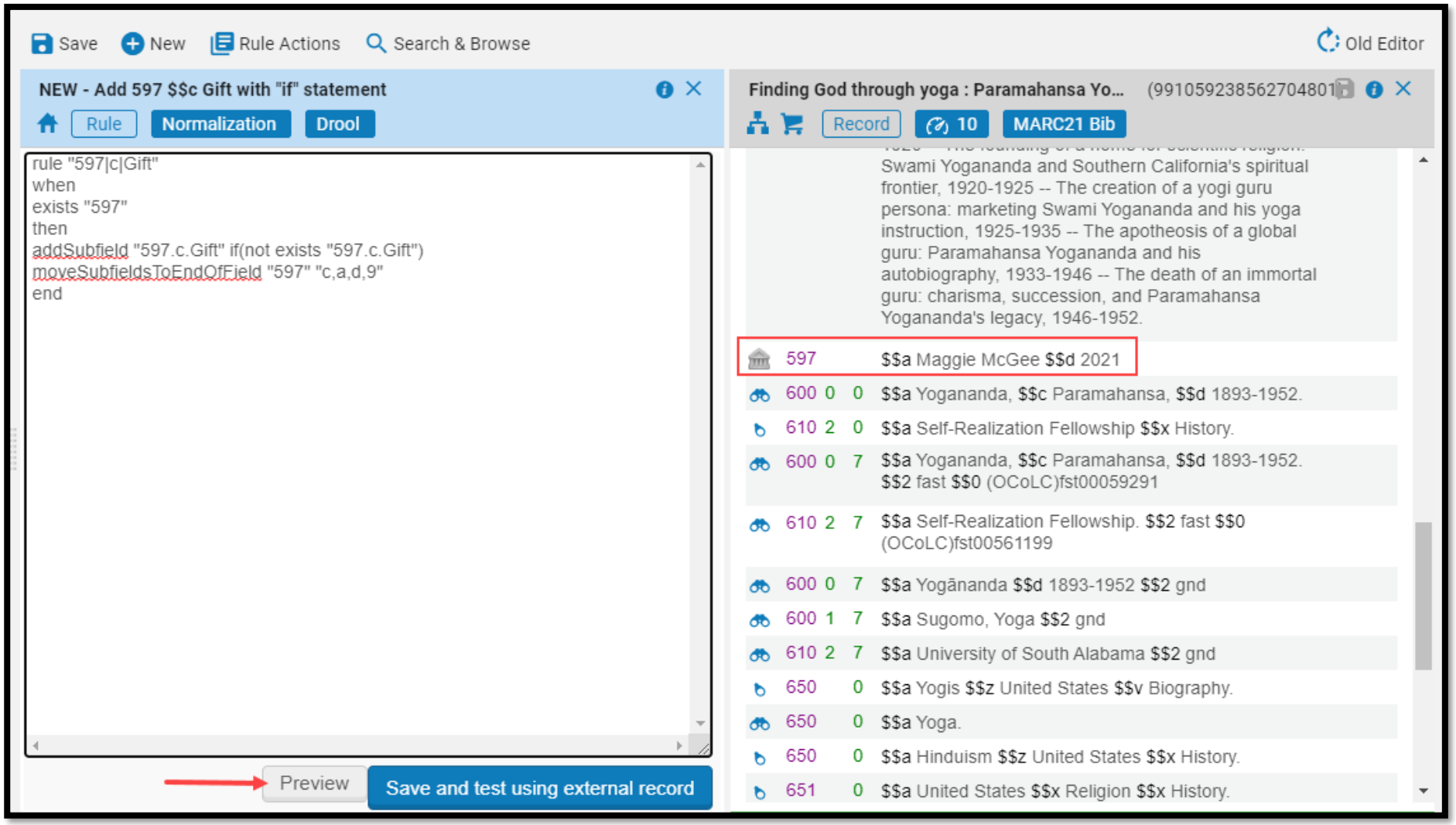




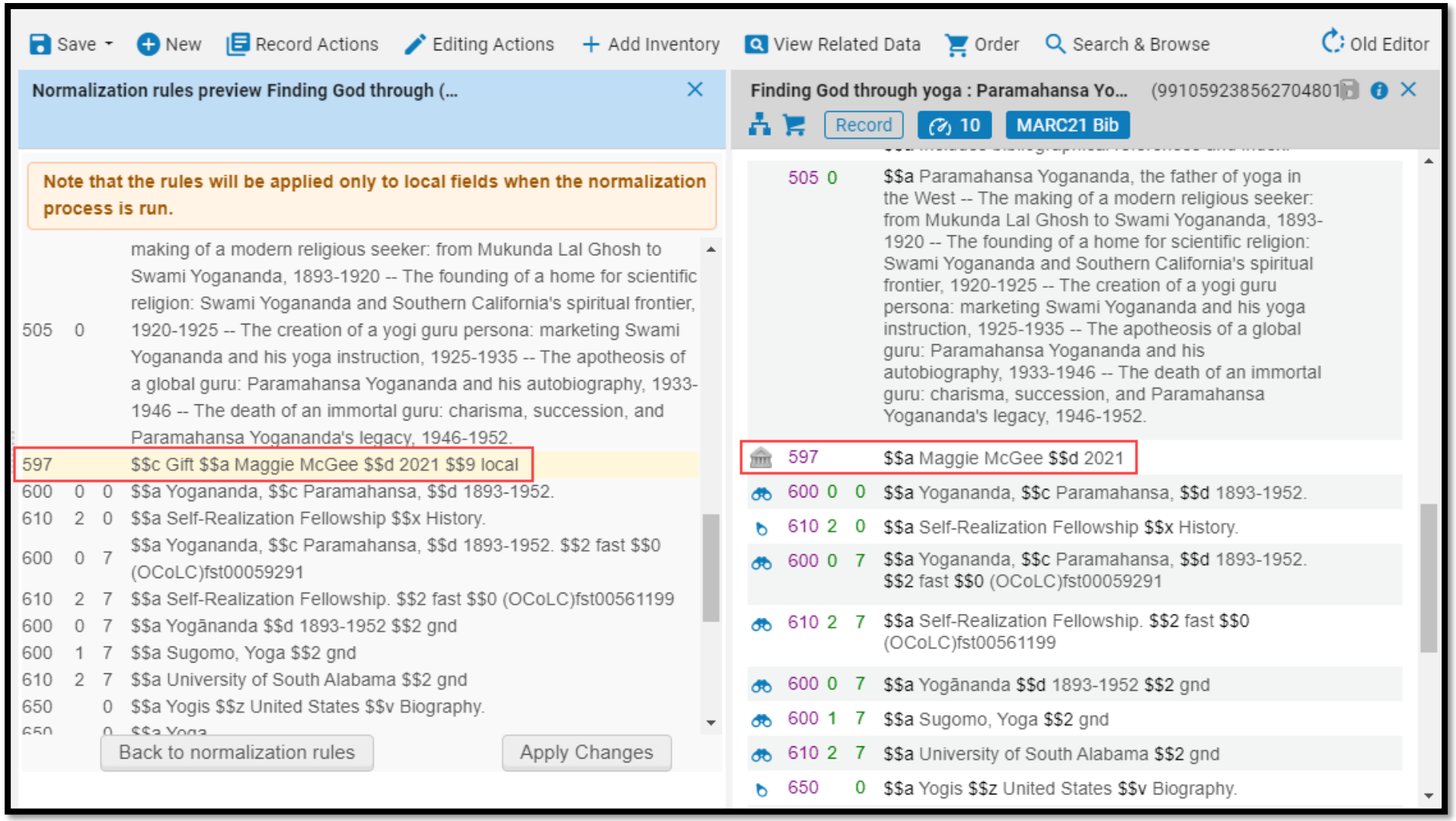
Testing a Norm Rule in the MD Editor

- With the norm rule open in the MD Editor:
  1. Click on the Editor split mode icon (F6)
  2. Click the **Records** tab
  3. Click on the **MARC21 Bib** record type
  4. Search for the bib record
  5. **Left-click** to open the bib record on the right-hand side of the main pane
  6. Click on the rule on the left-side of the main pane
  7. Click ***Preview***
     - A preview of the rule changes will display
     - Click Back to Normalization Rule
  8. Edit, preview, and save the rule if the rule did not work as expected in step 9
  9. Click "***X***" to close the indication rule
  10. Release the bib record (***Records Actions>Release Record*** / Alt+Shft+R)

Testing a Norm Rule in the MD Editor



Testing a Norm Rule in the MD Editor



Try Creating and Testing a Norm Rule

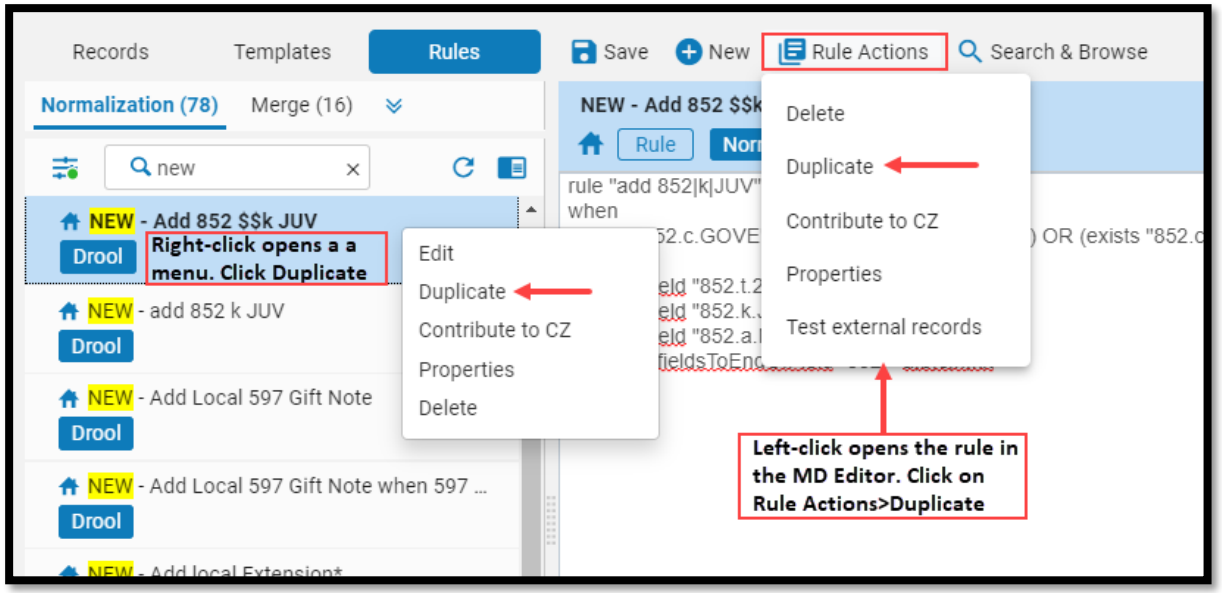
1. Try creating a bib nor rule with the following criteria for:
    1. Condition/Element with an “if” statement
        1. Add $$d 2021 to 597 when:
            1. 597 exists
            2. $$d 2021 does not exist
        2. Reorder subfields using moveSubfieldsToEndOfField
    2. TRUE condition with an “if” statement
        1. Add $$d 2021 to 597 when:
        2. $$d 2021 does not exist or
        3. if exists 597
2. Test the norm rules in the MD Editor with a bib record that:
    1. Has a local MARC 597
    2. Does not have a local MARC 597

DUPLICATING, EDITING, & TESTING NORMALIZATION RULES

SUNY The State University of New York

Duplicating an Existing Norm Rule

- Norm rules should only be edited by the user who created the rule
- DO NOT edit rules created by other users
- DO NOT edit rules that begin with SUNY
- Norm rules can be duplicated by left or right clicking on the rule name
- “Placement of new rules” is applied when duplicating a rule



Duplicating an Existing Norm Rule

- To duplicate a norm rule:
  1. Open the MD Editor
  2. Check the “Placement of new rules” is local
  3. Click on the Rules tab
  4. Click on the **Normalization** rule type
  5. Search for the norm rule
  6. Left-click on the rule name
  7. The Normalization Rules screen opens
     1. Edit the name using the SUNY naming convention
     2. Edit the description
     3. Keep Shared and Enabled checked
     4. Click ***Save***
  8. Edit the rule to meet the user’s needs if necessary
  9. Save the Changes
  10. Test the Rule

SUPPORTING DOCUMENTATION

SUNY The State University of New York

Supporting Documentation

- SUNY
  - SUNY Normalization Rules LibGuide

- Ex Libris
  - Working with Normalization Rules
  - Normalization Rules Examples