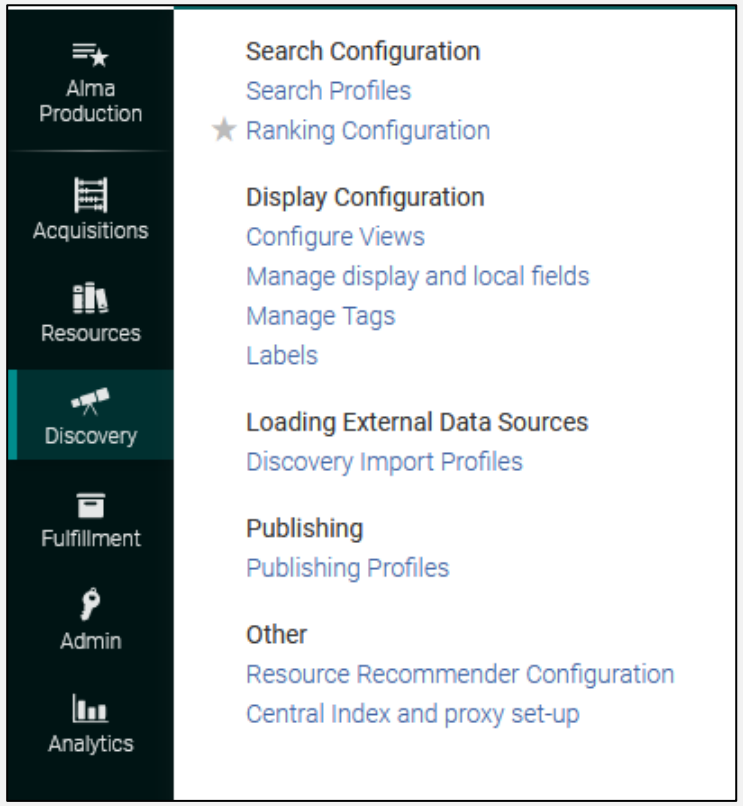

Ex Libris
Part of Clarivate
IGeLU
International Group of Ex Libris Users
Primo VE Basic Normalization Rules Workshop
BenYishai Ze'evi | September 14, 2023

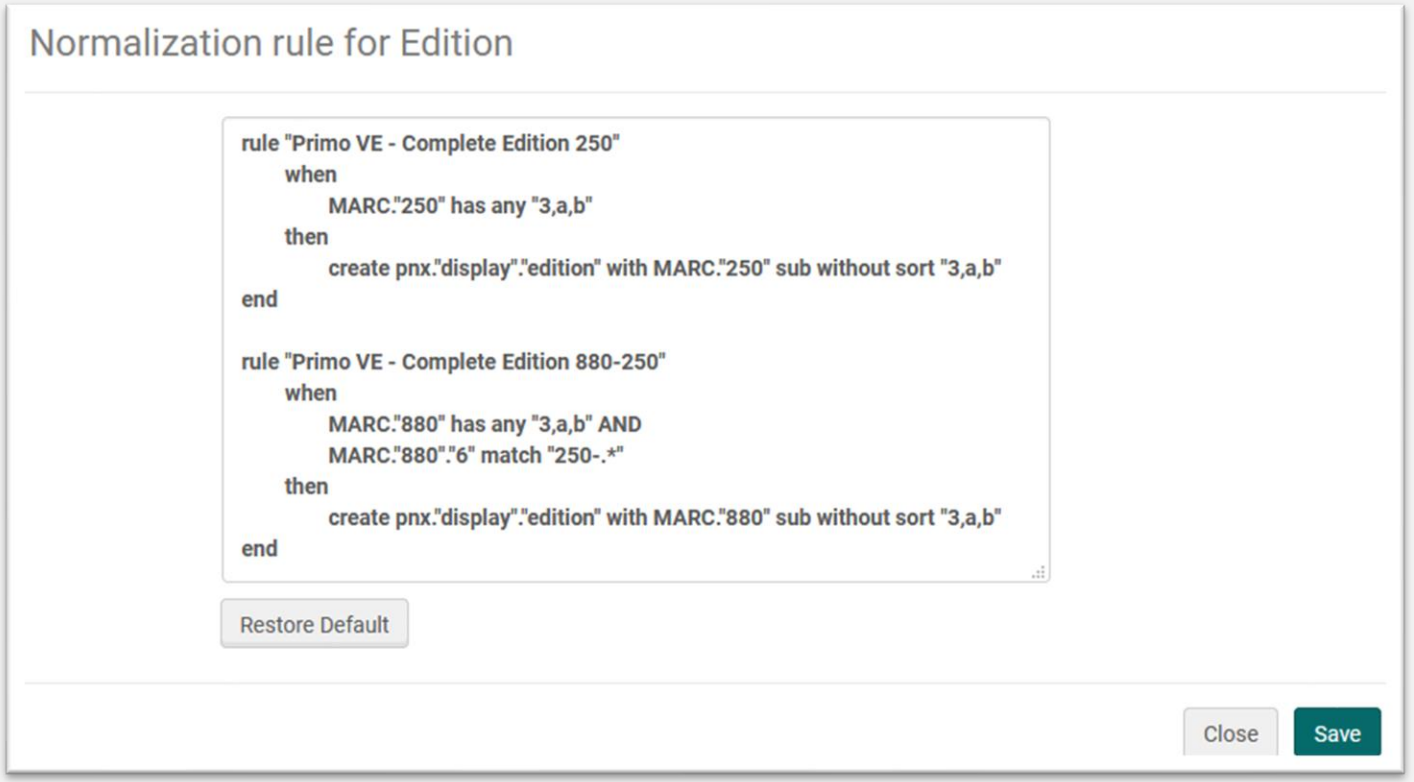
- Normalization Structure & Syntax
- Display Fields
- Local Fields

Mapping to the Display, Facets, and Search Sections in the Primo VE Record



DROOLS

```
rule "title"
     when
          Condition1 <Boolean operator>
          Condition2 <Boolean operator>
              ...
          Condition(n)
     then
          Action1
          Action2
          Action(n)
end
```



Documentation & Examples

Home » Primo » Product Documentation » Primo VE » Primo VE (English) » Display Configuration » Configuring Normalization Rules for Dis

Configuring Normalization Rules for Display and Local Fields

Home » Primo » Product Documentation » Primo VE » Primo VE (English) » Display Configuration » Managing Display and Local Fields for Pı

Managing Display and Local Fields for Primo VE

Home » Primo » Product Documentation » Primo VE » Primo VE (English) » Other Configurations » Configuring Local Search and Facet Field

Configuring Local Search and Facet Fields for Primo VE

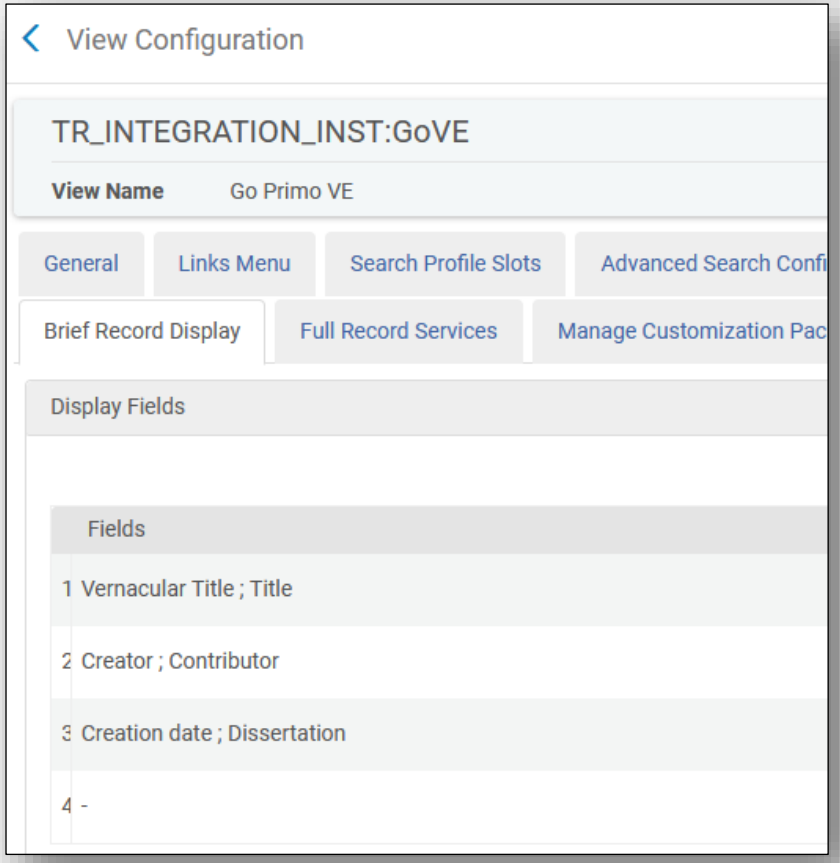
Home » Primo » Product Documentation » Go VE » Step 3: Configuration » Go VE: Converting Local Fields in Primo to Primo VE

Go VE: Converting Local Fields in Primo to Primo VE

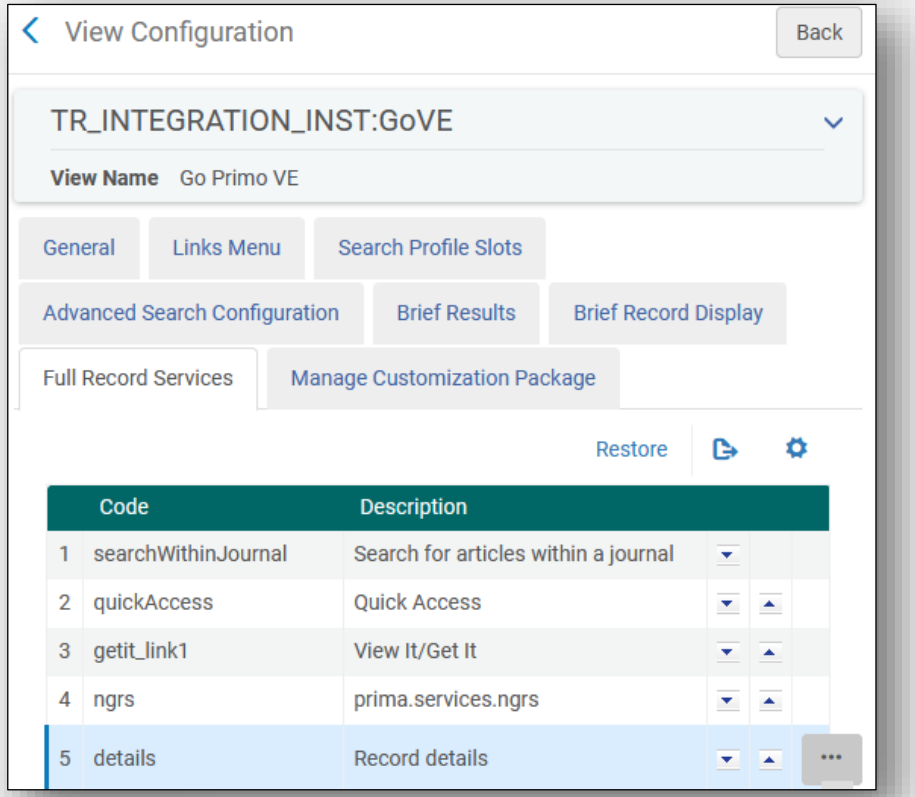
Home » Primo » Product Documentation » Go VE » Step 3: Configuration » Go VE: Customizing the Default Display Fields in Primo VE

Go VE: Customizing the Default Display Fields in Primo VE

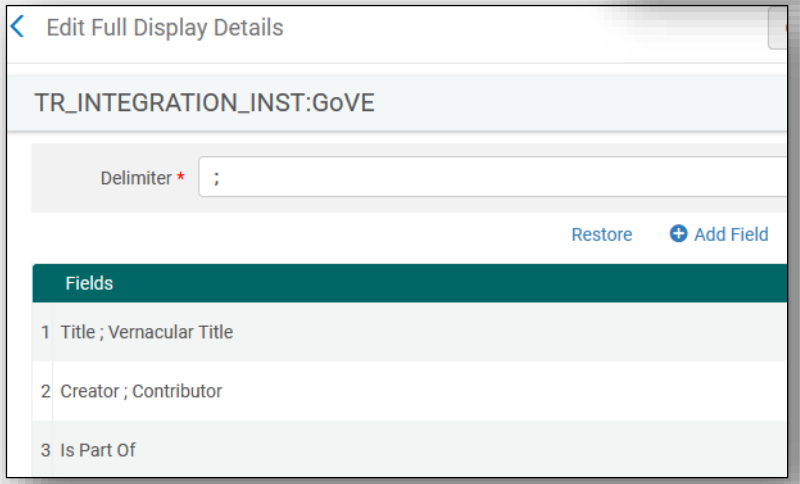
Display Fields



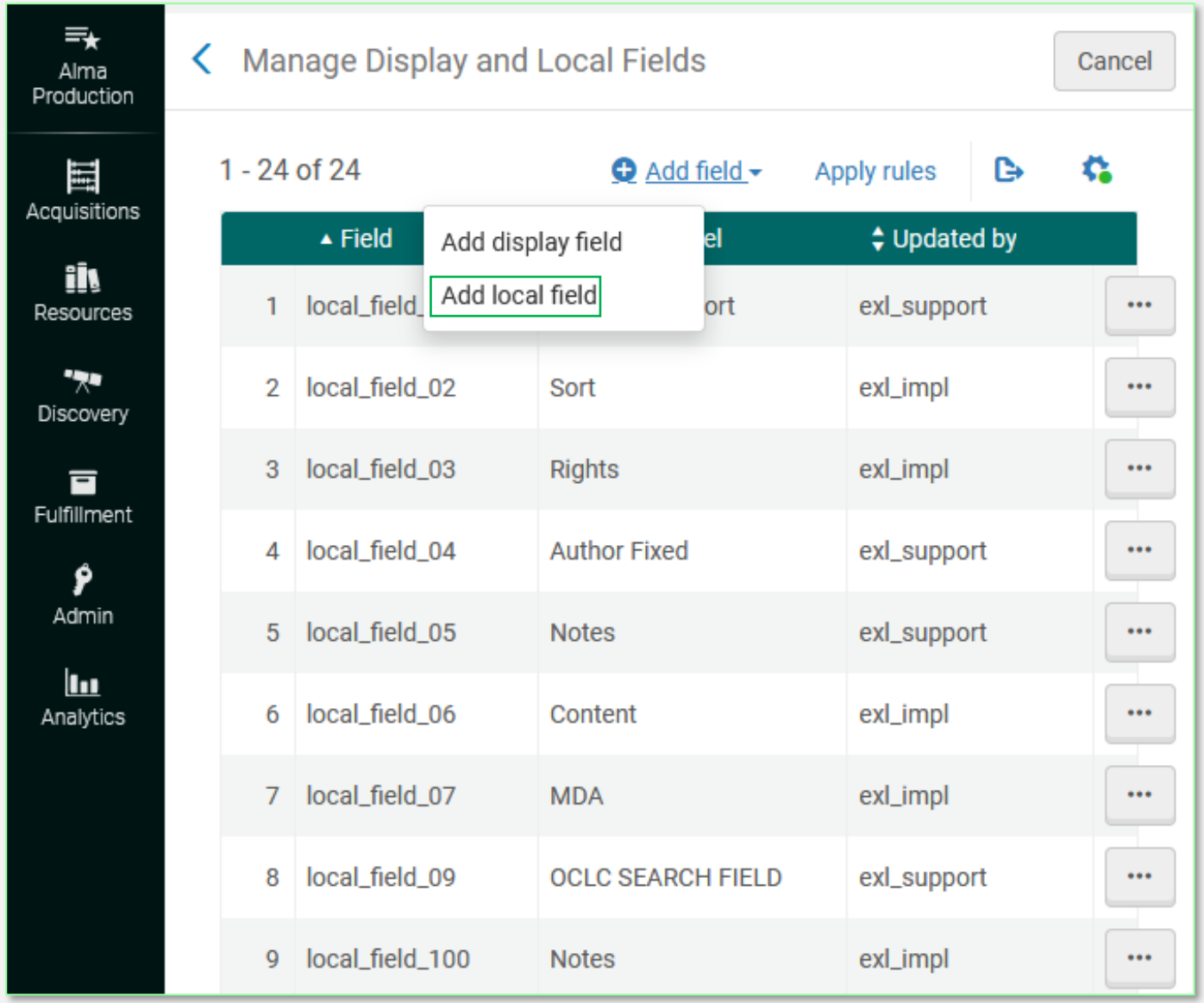
Brief Display



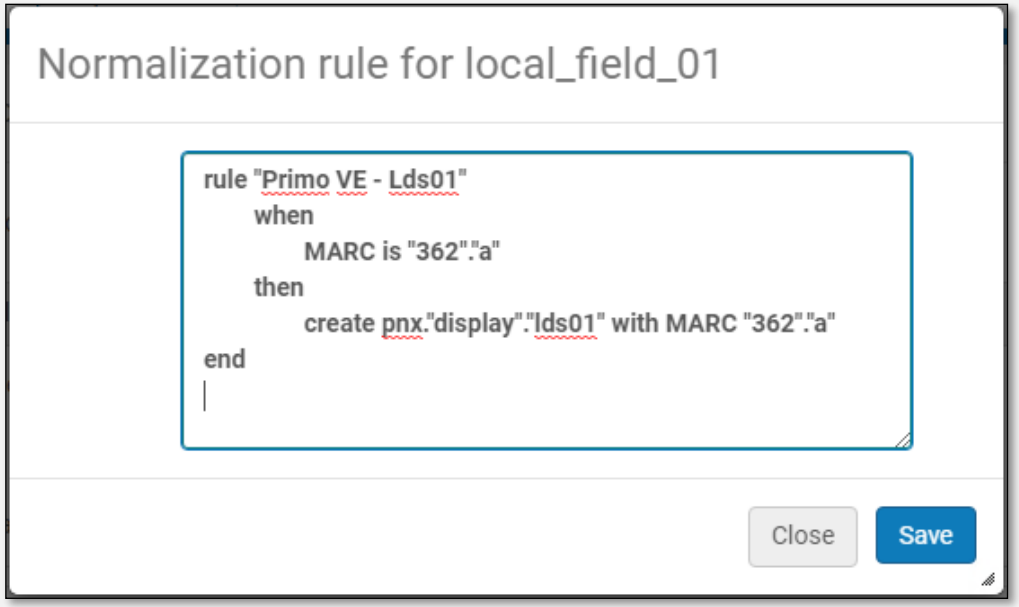
Full Display

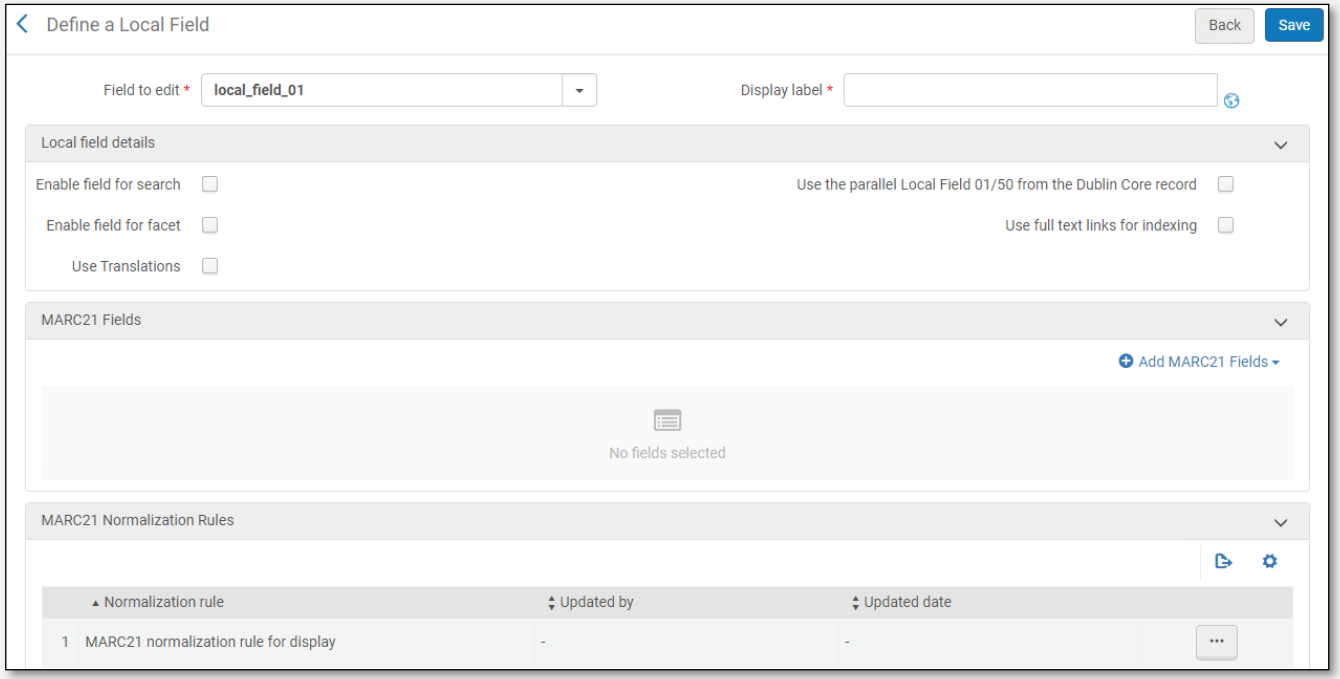


Local Fields



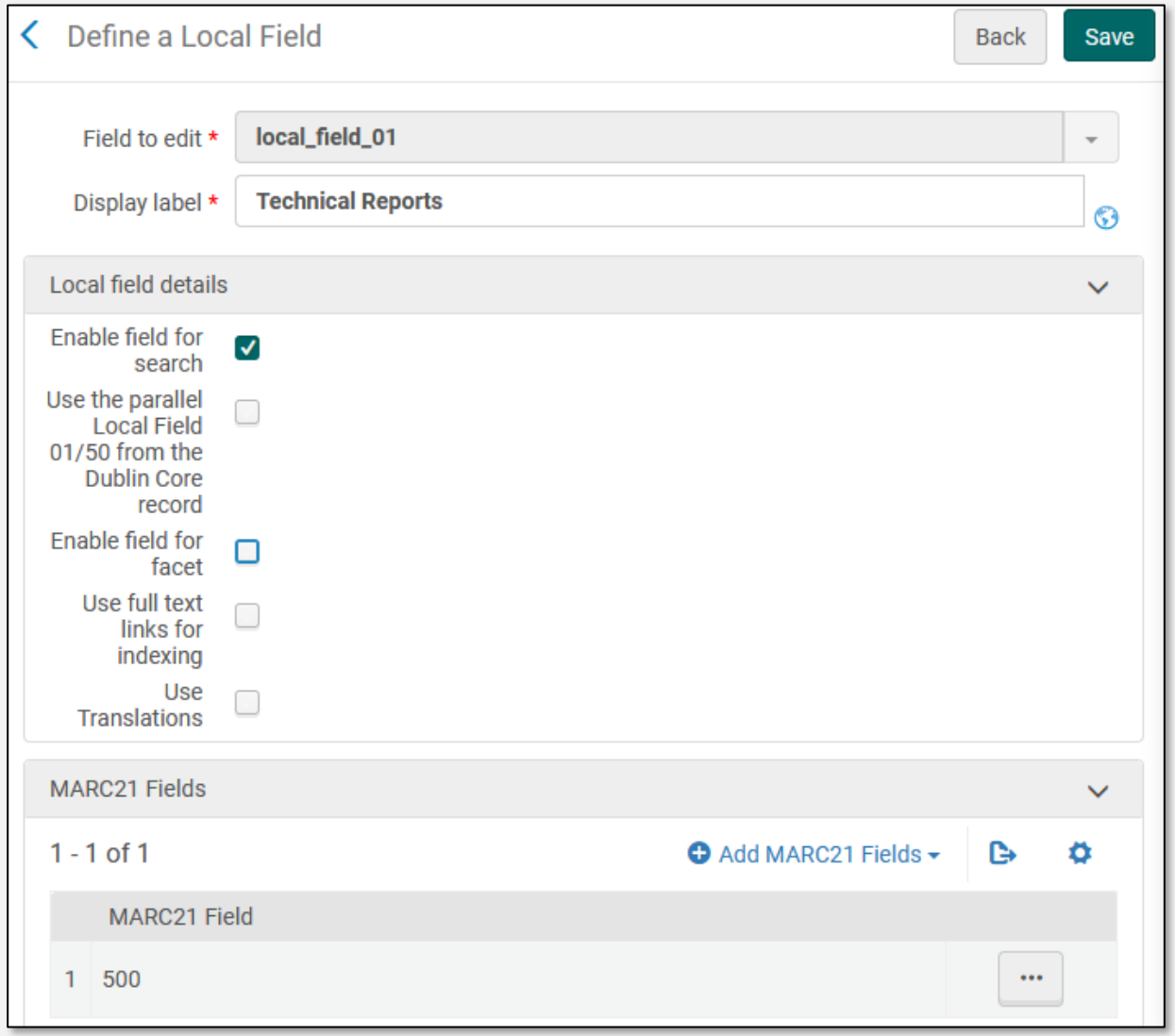
Local Fields





MARC to local_field

- The available fields for **MARC21** and **KORMARC** are: 009, 09X, 490, 5XX, 69X, 9XX
- The available fields for **UNIMARC** are: 3XX, 6X9, 69X, 9XX
- The available fields for **CNMARC** are: 009, 225, 3XX, 410, 411, 69X, 9XX



Rule Syntax

```
Title
When
    • Existence checks
    • additional conditions
Then
    …
End
```

Existence Checks

When

- `MARC is "<field>"`
- `MARC is "<field>"."<subfield>"`
- `MARC."<field>" has any "<subfield list>"`

• **is**
Applies the actions when a specific field/subfield exists
	`MARC is "880"`
	`MARC is "260"."c"`
	`MARC.control is "001"`

• **match** contents of subfield matches a regular expression	`MARC."880"."6" match "505-.*"`
• **has any** field exists and has any of the specified subfields	`MARC."700" has any "a,b,c,d,e,j,q,u"`
• **equals** entire contents of the specified subfield matches	`MARC."260".ind"1"  equals "3"`

Rule Syntax

Title
When
…
Then

- source definition
- Transformation actions
- Assignment command

End

Assignment Actions

Then

- `set <PNX field> to <field>`
- `create <PNX field> with <field>`

String Manipulation

Actions – String manipulation

TEMP 1 = *A person's a person,*
TEMP 2 = *no matter how small.*

- **remove substring using regex**
  removes a substring from the contents of a temporary field using a specified regular expression

```
remove substring using regex (TEMP"1","(/|:|;|=|,)+$")
```
TEMP 1 = A person's a person

- **add prefix**
  Adds the specified value to the beginning of the temporary field

```
add prefix (TEMP"1","Dr. Seuss: ")
```
TEMP 1 = Dr. Seuss: A person's a person,

- **concatenate with delimiter**
  concatenates the contents of two temporary fields and separates them with a delimiter
  The result is placed in the first temporary field

```
concatenate with delimiter (TEMP"1",TEMP"2"," ")
```
TEMP 1 = A person's a person, no matter how small.

- **replace string by string**
  replaces all occurrences of a regular expression with a specified value

```
replace string by string (TEMP"1","[^a-z]","\\?")
```
TEMP 1 = personsaperson

Actions – String manipulation

TEMP 1 = *A person's a person,*
TEMP 2 = *no matter how small.*

- **return list using regex**
  searches for all occurrences of a regular expression in the second temporary field and copies them as a list to the first temporary field

```
return list using regex (TEMP"1",TEMP"2","[a-z]{3}")
```

- **remove string**
  removes a specified string from a temporary field

```
remove string (TEMP"1","person")
```
TEMP 1 = a 's a ,

- **remove leading and trailing spaces**
  removes beginning and trailing spaces in a temporary field.

```
remove leading and trailing spaces (TEMP"1")
```

- **replace spaces**
  replaces all spaces in a temporary field with a specified character or string

```
replace spaces (TEMP"1","+")
```
TEMP 1 = A+person's+a+person,

Example

Actions – String Manipulation

add prefix

```
when
        MARC is "245"."a"
then
set TEMP"1" to MARC."245" subfields "a"
add prefix(TEMP"1","J. K. Rowling : ")
```

245 |a Harry Potter

TEMP"1" = J. K. Rowling : Harry Potter

Example

Actions – String Manipulation

concatenate with delimiter

```
When
      MARC.“650" has any "a,b,c,v"
Then
      set TEMP”1” to MARC.“650" subfields “a"
      set TEMP”2” to MARC.“650" subfields “v"
      concatenate with delimiter (TEMP"1",TEMP"2",“---")
```

650 |a Astronomy |v Juvenile literature.

TEMP”1” = Astronomy --- Juvenile literature.

Example

Actions – String Manipulation

delimited by

```
When
        MARC.“650" has any "a,b,c,v"
Then
set TEMP“1" to MARC."650" subfields “a,v" delimited by " --- "
```

650 |a Astronomy |v Juvenile literature.

TEMP”1” = Astronomy --- Juvenile literature.

Example

Actions – String Manipulation

replace string by string

When

```
MARC is "999"."a"
```

Then

```
set TEMP"1" to MARC."999" subfields "a"
replace string by string (TEMP"1","^mailto:","Email: ")
```

999 |a mailto:John.Doe@exlibrisgroup.com
TEMP"1" = Email: John.Doe@exlibrisgroup.com

Example

Actions – String Manipulation

remove string

When

```
MARC is "999"."a"
```

Then

```
set TEMP"1" to MARC."999" subfields "a"
remove string (TEMP"1","mailto:")
```

999 |a mailto:John.Doe@exlibrisgroup.com

TEMP"1" = John.Doe@exlibrisgroup.com

Example

Actions – String Manipulation

remove substring using regex

When

```
MARC is "245"."a"
```

Then

```
set TEMP"1" to MARC."245" subfields "a"
remove substring using regex (TEMP"1","(/|\.|;|=|,)+$")
```

245 |a Harry Potter.

TEMP"1" = Harry Potter

Q: how do I display MARC 500 based on the condition that MARC 035 begins with: (OCLC) ?

```
rule "Primo VE - Lds02"
when
       MARC.“500" has any "a,z" AND
       MARC is "035"."a" AND
       MARC."035"."a" match "\(OCLC\).*"
then
 set pnx."display"."lds02" to MARC.“500" sub without sort "a,z"
end
```

Thank you!

BenYishai Ze'evi

Ben-Yishai.Zeevi@Clarivate.com

We hope you enjoyed our presentation as much as we did!

Please fill out this short survey:

https://www.surveymonkey.com/r/WVLHJGG

About Clarivate

Clarivate is the leading global information services provider. We connect people and organizations to intelligence they can trust to transform their perspective, their work and our world. Our subscription and technology-based solutions are coupled with deep domain expertise and cover the areas of Academia & Government, Life Sciences & Healthcare and Intellectual Property. For more information, please visit clarivate.com

© 2023 Clarivate

Clarivate and its logo, as well as all other trademarks used herein are trademarks of their respective owners and used under license.