



Custom Attribute Map rules

Last Modified on 04/18/2023 10:49 am EDT

Sometimes, a [direct reader attribute mapping](#) is not possible.

We generally see this situation arise with:

- Groups, either because the IdP doesn't have groups, or the groups the IdP has are not relevant to KnowledgeOwl.
- Custom fields that have a different value/meaning in your IdP and need to be "translated" to a different value in KnowledgeOwl.

In these situations, it can be useful to define a rule that will look at some attribute coming from the IdP and then update a KnowledgeOwl field based on that rule. We call these Custom Attribute Map Rules [CAMR].

There are two things to keep in mind when working with CAMRs:

- **Direct reader attribute mappings always overwrite custom attribute mappings.**
This means that you can't, for example, have a direct mapping for the Reader Groups field and also have a custom attribute map rule that sets the Reader Groups field. The direct mapping will always override the CAMR. As a general rule of thumb, if you're going to set a field using a CAMR, it's best to leave the direct mapping for it blank.
- **At this time, only one rule can be applied to a specific reader/attribute combination.**
If you have five rules for what happens with the groups field, only the first rule that is a match will be applied for a given reader, even if a reader technically matches all 5 rules. Generally, the oldest rule is the one which is applied first.

We offer three types of CAMRs:

1. **Default:** The simplest of rules, the default rule will check to see if a specified IdP attribute is present at all, and will set a reader field when it is. It does not care what the actual value of the IdP attribute is.
2. **Exact matches:** These will look for an exact match for an attribute value coming from the IdP and set a reader field only when it finds that exact match.
3. **Regex pattern:** The most flexible and complex rule, in these you can define a regular expression (regex) pattern to match an IdP value, and set a reader field accordingly. We use the PHP flavor of regex.

Default rules


Default rules are our simplest rule. The rule basically says: if an incoming IdP attribute has *any* value, set the selected KnowledgeOwl field for the reader.

It does not matter what the value of the attribute actually is.

You can only set one default rule per IdP attribute name (so, for example, you can't have two default rules that both checked the custom1 attribute coming from your IdP).

Default rules will show a rule logic of **IS EQUAL TO** * (anything) in the display:

Reader field: Reader Groups
IdP attribute: uid
Rule logic:
IF
uid
IS EQUAL TO
* (anything)
SET
Reader Groups
TO

 Production  Support

Use case

Default rules are a fantastic choice if you want to assign all readers to one or more reader groups, or set one of the custom fields to be the same value for all SSO readers.

Setup

To create a default rule:

1. Go to **Settings > SSO**.
2. Open the **SAML Attribute Map** tab.
3. Click the **+ Create New Rule** button in the **Custom Attribute Map Rules** section. This will open a pop-up where you can define the rule:

Create Advanced SAML Map Rule

IdP attribute name

4

IdP attribute value matching type

Exact matches ?

Regex pattern ?

5 Default ?

*Only one default is allowed per IdP attribute name

Reader attribute

6 Reader First Name

Reader attribute value

7

[Cancel](#) [Create Rule](#)

4. Add the **IdP attribute name** you'd like to base the rule on. This should be formatted exactly as it appears in the IdP.

- If you're not sure of the attribute name, you can review your IdP settings or **Enable debug mode** in the **SAML Settings** tab and try to login--you should see a full list of the attributes coming from the IdP.
- For a default rule you'd like applied to all readers coming in through SSO, use any attribute that is always populated from your IdP. Email address, username or user ID, first name, or last name are often good candidates.

5. Select **Default** as the IdP attribute value matching type.

6. Use the **Reader attribute** dropdown to select the reader field you'd like to update based on this rule.

7. Add the **Reader attribute value** you'd like that reader attribute set to.

8. Click **Create Rule**.

Once created, you'll be able to see, edit, and delete the rule in the **Custom Attribute Map Rules** section.

Example

To set all of your incoming SSO readers to a single group, you can match against an IdP attribute which is always populated, such as the userID or email address.

Here, we have left the Reader Groups field mapping empty in the direct attribute mapping list:

SSO ID	<input type="text" value="uid"/>	<small>This field is required. Email address is allowed.</small>
Username / Email	<input type="text" value="email"/>	<small>This field is required.</small>
First Name	<input type="text" value="firstName"/>	
Last Name	<input type="text" value="lastName"/>	
User Icon	<input type="text"/>	<small>Expecting URL to image.</small>
Reader Groups	<input type="text"/>	<small>Can use a comma separated list or an array to assign multiple reader groups.</small>
Custom Field 1	<input type="text" value="custom1"/>	
Custom Field 2	<input type="text"/>	

And we've set up a default rule using our incoming user ID field, which is called uid in our IdP and cannot be empty, to assign all readers to the Production and Support reader groups:

Create Advanced SAML Map Rule
✕

IdP attribute name

IdP attribute value matching type

Exact matches ?
 Regex pattern ?
 Default ?

*Only one default is allowed per IdP attribute name

Reader attribute

Reader Groups
▼

Assign to reader groups

🏠 Production ✕

🏠 Support ✕

[Cancel](#)
Create Rule

Once saved, this rule logic displays like this:

Reader field: Reader Groups

IdP attribute: uid

Rule logic:

IF

uid


IS EQUAL TO


* (anything)

SET

Reader Groups

TO

 Production

 Support

With this rule in place, every reader who logs in will be assigned to the Production and Support Reader Groups.

Exact matches rules

With exact match rules, you define the incoming IdP attribute the rule should examine as well as a specific value it should match. If the exact value is matched when a reader authenticates, it will set a reader field based on the rule.



You should not use exact match rules when an incoming attribute is an array (like a list of groups). We collapse array attributes into a single field. For attributes containing arrays, use a regex rule to match within the array.

Default rules will show a rule logic of **IS EQUAL TO** *value* in the display; if you add multiple exact matches, the **IS EQUAL TO** will show the exact matches as an **OR** list in square brackets:

Reader field: Custom Field 1

IdP attribute: country

Rule logic:

IF

country

IS EQUAL TO

[US OR USA OR United States of America]

SET

Custom Field 1

TO

United States

Use case

The exact match rule is useful when you have a specific attribute value in your IdP that you'd like to transform into a

different value in KnowledgeOwl.

For example, maybe your company operates in five offices in three countries, and your IdP captures an office attribute for each employee.

But in your knowledge base, you don't care about offices. You have a reader group for each country only.

You can set up an exact match rule for the offices of each country to map your readers to those groups:

- office attributes: Colorado or New York > US employees reader group
- office attributes: London or Manchester > UK employees reader group
- office attribute: Melbourne > Australia employees reader group



Exact matches are true exact matches; if you need to do a fuzzier or more flexible match, use a regex rule.

Setup

To create an exact match rule:

1. Go to **Settings > SSO**.
2. Open the **SAML Attribute Map** tab.
3. Click the **+ Create New Rule** button in the **Custom Attribute Map Rules** section. This will open a pop-up where you can define the rule:

Create Advanced SAML Map Rule ×

IdP attribute name

4

IdP attribute value matching type

5 Exact matches ?

Regex pattern ?

Default ?

*Only one default is allowed per IdP attribute name

IdP exact match value

6

7

Reader attribute

8

Reader attribute value

9

[Cancel](#)

4. Add the **IdP attribute name** you'd like to base the rule on. This should be formatted exactly as it appears in the **IdP**.
 - If you're not sure of the attribute name, you can review your **IdP settings** or **Enable debug mode** in the **SAML Settings** tab and try to login--you should see a full list of the attributes coming from the **IdP**.
5. Select **Exact matches** as the **IdP attribute value matching type** (this is usually selected by default).
6. Add the exact value of of the incoming attribute you'd like to match against as the **IdP exact match value**.
7. If you'd like to look for multiple values to match against, click the + **Add value** button to add additional values for this rule.
 - Multiple values will be treated as **OR values**--the exact match will match for the first value **OR** the second value **OR** the third value, etc.
8. Use the **Reader attribute** dropdown to select the reader field you'd like to update based on this rule.
9. Add the **Reader attribute value** you'd like that reader attribute set to.
10. Click **Create Rule**.

Once created, you'll be able to see, edit, and delete the rule in the **Custom Attribute Map Rules** section.

Example

Let's say our company operates in five offices:

- Denver, Colorado
- New York, New York
- London, England
- Manchester, England
- Melbourne, Australia

We've set up reader groups in our knowledge base for each country:

- US employees
- UK employees
- AUS employees

An employee can only belong to one office, and our **IdP** tracks this information for each employee in an "office" field.

We want to set up a rule that will check that office field and then assign readers to the appropriate country employee group.

We won't use direct mappings for our groups field:

SSO ID
This field is required. Email address is allowed.

Username / Email
This field is required.

First Name

Last Name

User Icon
Expecting URL to image.

Reader Groups
Can use a comma separated list or an array to assign multiple reader groups.

Custom Field 1

Custom Field 2

And we'll create three different exact match rules, one for each country. Here's how I set up the US employees rule (note that we used the + Add value button so we have two exact matches in the list, since we have two offices in the US):

Create Advanced SAML Map Rule ✕

IdP attribute name

IdP attribute value matching type

Exact matches ?

Regex pattern ?

Default ?

*Only one default is allowed per IdP attribute name

IdP exact match value

IdP exact match value 🗑

Reader attribute

Assign to reader groups

[Cancel](#)

Here's the final configuration for the rules all three countries:

Reader field: Reader Groups

IdP attribute: office

Rule logic:

IF

office


IS EQUAL TO

Melbourne

SET

Reader Groups

TO

 AUS employees

Exact match for our Australia office

Reader field: Reader Groups

IdP attribute: office

Rule logic:

IF

office


IS EQUAL TO

[London **OR** Manchester]

SET

Reader Groups

TO

 UK employees

Exact matches for our two UK offices

Reader field: Reader Groups

IdP attribute: office

Rule logic:

IF

office


IS EQUAL TO

[Denver **OR** New York]

SET

Reader Groups

TO

 US employees

Exact matches for our two US offices

With this rule in place, every reader who logs in with an office in Melbourne will be assigned to the AUS employees

reader group; every reader who logs in with an office of either London or Manchester will be assigned to the UK employees reader group, and every reader who signs in with an office of either Denver or New York will be assigned to the US employees reader group.

Regex pattern rules

Regex rules are the most complicated, but also the most flexible. Regex is a common abbreviation of 'regular expressions'. Regular expressions are "a sequence of characters that specifies a *search pattern*. Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation." (from [Wikipedia - Regular expression](#)).

For regex Custom Attribute Mapping Rules, you define a regex pattern you'd like to search for, and then define the KnowledgeOwl field and value you'd like to set when that pattern is found.

We recommend only using these rules when neither of the other custom rule types will meet your needs, as they are a bit harder to write and more intimidating if you're not familiar with regex. If an attribute you'd like to use is an array containing a list of values, you must use regex to properly search and match within that list.



[regex101](#) is a handy site that allows you to try out regex rules against example words and phrases. When testing regex for use in KnowledgeOwl, select **PCRE2 [PHP >= 7.3]** under **FLAVOR**.

Regex pattern rules will show a rule logic of **MATCHES** [pattern] in the display:

Reader field: Custom Field 2

IdP attribute: custom1

Rule logic:

IF

custom1

MATCHES

/Brie/i

SET

Custom Field 2

TO

Cheese lover

Use case

Regex pattern rules are useful when there are patterns within one of your IdP attributes that doing exact matches for might be tedious. For example, if all our internal IdP groups are prefixed with **KO_** and all of our contractor or customer IdP groups are prefixed with **EXT_**, and we really only care about this KO/not-KO distinction, a regex rule makes more sense than generating very large exact match rule lists.

See [Auto-Assign Groups By Email Rules \(SSO Edition\)](#) for a walkthrough on how to set up a rule to auto-assign everyone at a given email domain to one or more reader groups.

Setup

To create an exact match rule:

1. Go to **Settings > SSO**.
2. Open the **SAML Attribute Map** tab.
3. Click the **+ Create New Rule** button in the **Custom Attribute Map Rules** section. This will open a pop-up where you can define the rule:

Create Advanced SAML Map Rule

IdP attribute name

4

IdP attribute value matching type

Exact matches ?

5 Regex pattern ?

Default ?

*Only one default is allowed per IdP attribute name

IdP value regex pattern

6 / /i

Reader attribute

7 Reader First Name

Reader attribute value

8

[Cancel](#) [Create Rule](#)

4. Add the **IdP attribute name** you'd like to base the rule on. This should be formatted exactly as it appears in the IdP.
 - o If you're not sure of the attribute name, you can review your IdP settings or **Enable debug mode** in the **SAML Settings** tab and try to login--you should see a full list of the attributes coming from the IdP.
5. Select **Regex pattern** as the IdP attribute value matching type.
6. Add the regex pattern you'd like to match that attribute value against as the **IdP value regex pattern**.
7. Use the **Reader attribute** dropdown to select the reader field you'd like to update based on this rule.
8. Add the **Reader attribute value** you'd like that reader attribute set to.

9. Click Create Rule.

Once created, you'll be able to see, edit, and delete the rule in the **Custom Attribute Map Rules** section.

Example

Let's say the KnowledgeOwl IdP is used for authenticating KnowledgeOwl employees as well as KnowledgeOwl customers. Our employees might be segmented into different groups, like:

- KO_Support
- KO_Security
- KO_Development

Whereas our customers might be segmented into groups based on subscription plan:

- EXT_Flex
- EXT_Business
- EXT_Enterprise

Within our knowledge base, we only segment content by two reader groups:

- Internal users (KO employees)
- External users (Customers)

No account belongs to both KO_ groups and EXT_ groups.

Since the two never overlap, we could create two regex pattern rules to look for these prefixes and assign our users to the appropriate reader group in our knowledge base.

We would not include groups in our direct mapping:

SSO ID	<input type="text" value="uid"/>
	<small>This field is required. Email address is allowed.</small>
Username / Email	<input type="text" value="email"/>
	<small>This field is required.</small>
First Name	<input type="text" value="firstName"/>
Last Name	<input type="text" value="lastName"/>
User Icon	<input type="text"/>
	<small>Expecting URL to image.</small>
Reader Groups	<input type="text"/>
	<small>Can use a comma separated list or an array to assign multiple reader groups.</small>
Custom Field 1	<input type="text" value="custom1"/>
Custom Field 2	<input type="text"/>

And we could create a regex rule that looks for each of our prefix strings: Here's the rule looking for our EXT_ prefix:

IdP attribute name

IdP attribute value matching type

- Exact matches [?](#)
- Regex pattern [?](#)
- Default [?](#)

*Only one default is allowed per IdP attribute name

IdP value regex pattern

 /i

Reader attribute

Assign to reader groups

 Start typing group name...[Cancel](#)[Update Rule](#)

Sample configuration for the EXT_ matching rule

Here's what the two rules look like:

Reader field: Reader Groups

IdP attribute: groups

Rule logic:

IF

groups

MATCHES

/KO_/i

SET

Reader Groups

TO

Regex pattern for our KO_ groups

Reader field: Reader Groups

IdP attribute: groups

Rule logic:

IF

groups


MATCHES

/EXT_/i

SET

Reader Groups

TO

 External users

Regex pattern for our EXT_groups

With these rules in place:

- Anyone with any IdP KO_ group would be assigned to the Internal users reader group
- Anyone with any IdP EXT_ group would be assigned to the External users reader group