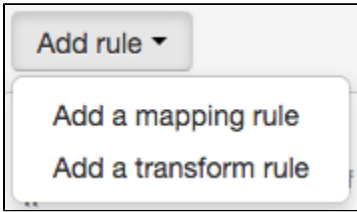# Attribute transformation

When you add an attribute mapping rule there is also an option to add an attribute transform rule.
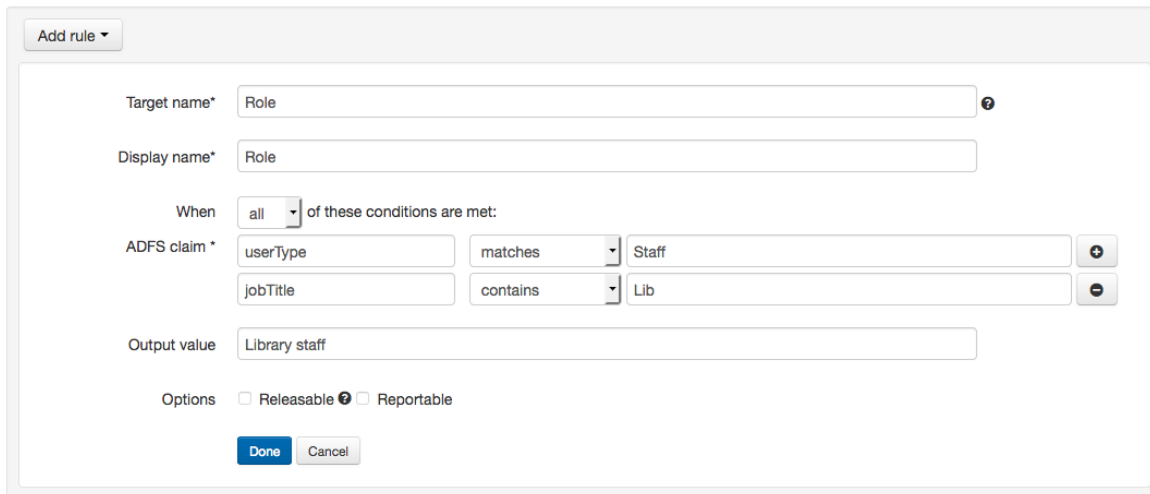

This is for situations such as:

- where the value of an attribute is correct, but not presentable
    - e.g. turn a value of `'CN=Filius Flitwick , DN=Teaching Staff, DN=ad, DN=hogworts, DN=sch, DN=uk'` into `'Professor'`
- where the value you need for an attribute is not available from your system, but you can infer it from other values that are
    - e.g. if `role` contains `Staff` *and* `house` has a value, then set `headOfHouse = true`.

You set up attribute transform rules in a similar way to how you map permission sets:

1. Go to Management > Connections and select the attributes tab of the connection you want to work with

2. Click the add rule button and select the transform rule option

3. The target name will suggest attributes that already exist in your schema. You will need to use one of these schema attributes if you want the data to be releasable to third parties or reportable under statistics.

4. The display name is how it will be presented in the interface when you look at an account

5. Define your rule or rules and whether it is any or all of them that need to be true to assign the output value

6. Specify an output value

7. Add any other rules and output values for that target

8. Click done and save

Anything to watch out for?

You can have only have one rule of any type per target attribute.

If more than one set of conditions matches, then the attribute will have more than one value. All values are passed if that attribute is released, and all values are used by statistics if it is reportable. Unfortunately the interface cannot display muliti-valued attributes on accounts at the moment.

You cannot use transformed values in other transformations.

> A multi-valued attribute is one that has several values... at the same time. It's not several values joined together into one value, but several distinct values that are grouped together - the techies would call it an array. An example might be if instead of having different fields for home, work and mobile telephone number, there was one field called telephone numbers that contained any and all phone numbers for that person.

```
<telephoneNumbers>
        <value: "+1 202-456-1414">
        <value: "+44 (0)1 811 8055">
        <value: "+1 636-555-3226">
</telephoneNumbers>
```

The most common multi-valued attribute that is likely to come up is the 'memberOf' attribute from Active directory (or Azure) which contains security and distribution groups.

Where this becomes important is that when a rule looks at a multi-valued attribute, then *all* of the values are treated as discrete when evaluated for match conditions and if *any* of the values meet the condition the rule returns 'true' and does the thing. This is great if you're doing a positive match (e. g. 'contains') because in general if any of the values match you want the rule to apply, however you need to be very careful with negative matches (e. g. 'does not contain') because if any of the values meet that condition then the rule also returns a match - e.g. if your rule is 'does not contain: Staff' and the attribute has three values of ['Full Time Staff', 'Physics Staff', 'Cat appreciation society'], it will return 'true' since one of the values does not contain Staff.

This is one of those computer logic things that isn't obvious at first, but makes sense when you think it through and, once understood, opens many possibilities. In purely practical terms though is is often best to stick with the positive matches such as 'contains' or 'matches' when you're working with multi-valued attributes.

Example

Scenario:

An organisation has offices in 20 cities around the world. The directory they have connected to OpenAthens can pass the name of the office (e.g. Abuja office), but not the country. All users have access to the same set of resources and are not separated by sub-organistaion or permission set.

Management want statistics reports by country.

Method:

Because we want to use this in reports, we first must set up a text attribute in the schema editor and make sure it is marked as reportable. This example will assume one has been set up with a target name of `country`.

Next we add some transform rules along the lines of: When any of these match (office contains Edinburgh, office contains Bristol) the output value is 'UK'.

Add conditions and outputs for each country, then click done and save.