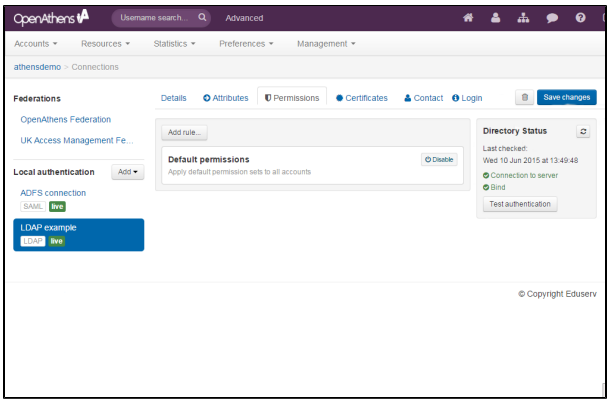
Permission set rules

This tab also has a function to suspend accounts base on rules.

Permission sets control which resources are accessible to your users and can be applied based on any attribute in your directory - for example if your directory has an attribute that identifies whether someone is staff or student in some way, you can use that difference to assign different permission sets and control access.

When you add a new connection, there will be one rule set up already to assign any default permission sets to your local accounts and this may be enough for your needs.



If you hover over the default permissions rule you will see an option to disable it if you do not want to assign default permission sets.

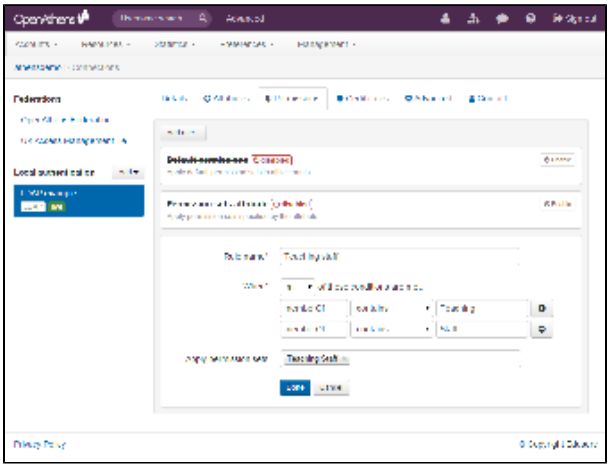
Additional ways to allocate permission sets

There are two other ways to specify permission sets: by rules and by attribute. Both can be used together, with and without also assigning default permission sets. Rules look at the values of attributes and make decisions based on those attributes, whilst a permission set attribute directly specifies the permission set name to use.

By rules

Before you start creating rules, you may like to discuss things with your IT team and get a list of the relevant attributes and typical values you will encounter from them. To add a rule:

1. Click the add rule button
2. Give it a name
3. In the 'When' area:
    a. Enter the name of the attribute in your local source. Attribute names are case sensitive.
    b. Select your condition, e.g. contains, starts with
    c. Enter your comparison
    d. If you need to add other conditions, add additional lines with the plus button and specify if any or all conditions must be met as appropriate. You can add as many lines as you need to.



4. Select the permission sets to apply. If you have a large number of sets you can start typing the name for a shorter list
5. When happy, click done and set up any other rules. You can create as many rules as you need to.
6. Use the save button when done and subsequent logins will use these new rules. (The save button is hidden whilst you are adding or editing rules)

Examples

When you want to assign a permission set when *memberOf* contains *Visitor* OR when *memberOf* contains *walk-in* you would specify both conditions and select when *any* conditions are met.

When you want to assign a permission set only when *memberOf* contains *Staff* AND when *memberOf* contains *Teaching* you would specify both conditions and select when *all* conditions are met.

Editing and removing rules

Move your mouse over the rule and select edit or remove.

Rules are removed as soon as you hit delete in the confirmation box, but editing (as with adding) requires you to save changes before they are applied.

Multi-value attributes and rules

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.

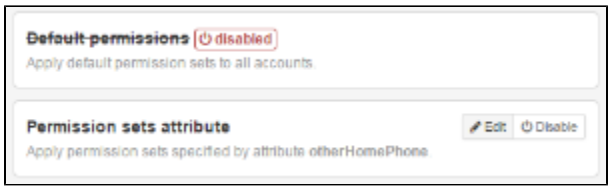
Where this becomes important is that when a rule looks at a multi-valued attribute, then *all* of the values are treated as discrete when evaluated for match conditions and if *any* of the values meet the condition the rule returns 'true' and does the thing. This is great if you're doing a positive match (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.

By attribute

To assign rules by attribute, you will need to store the permission sets names (e.g. abc#students) against an attribute in your directory and then specify that attribute in the interface. This option is not available if you are mapping users to organisations.

The permission sets attribute is disabled by default, so you must first enable it. You can then use the edit button to specify the attribute:



On LDAP connections there is the usual typeahead to help you find the attribute. For ADFS connections you must again enter the claim name exactly.

Multi-valued attributes can be very useful here as they allow you to pass multiple permission set names and all values passed will be assigned. In Active Directory, the rarely used 'other' attributes may be useful (e.g. otherFacsimileTelephoneNumber, otherPager, etc).

Anything to watch out for?

Expired permission sets will still be assigned if they match any rule or are specified by attribute. It is recommended that you not set expiry dates on permission sets.

When passing permission set names by attribute: if you pass a permission set name that does not exist in the same organisation as the user, they will not be applied and access may be affected. If you have an LDAP connection you can see these values by using the test mappings function.