# Manual attribute releases with ADFS Toolkit

## General

There are several reasons why a SP could need more attributes than provided in the Entity Categories provided

- The SP don't live up to the demands for a specific Entity Category but need the attributes to work
- The SP works with the provided attributes from the Entity Category but will 'look better' with added attributes
- The SP has incorrect metadata in the feed and needs other attributes than provided from the Entity Category
- The SP don't have any Entity Category but need more attributes than transient-id
- The SP is local for the institution so the attribute release can be done more freely

To fix this ADFSToolkit allow the IdP administrator to make manual attribute releases for specific SP's.

The manual releases are *added* to the release from the Entity Categories, if any. If the same attribute is released in a Entity Category and also in a manual release the *manual release will take over.*

## How to make a manual attribute releases for SP's

In the /config folder you will find a PowerShell script with the name get-ADFSTkManualSPSettings.ps1.

This script contains a function that should contain all specific overrides for attribute releases for a given entity.

For a given entity, we:

- create an empty TransformRules Hashtable
- assign specific transform rules that have a corelating TransformRules Object
- when complete, we insert the Ordered Hashtable transform into the Hashtable we return
- We can also get clever and inject a transform rule into the hashtable rather than reference an existing one

The TransformRules Hashtable is provided in the top of the script and that Hashtable is returned in the end. It looks like this:

```
# Hashtable that we will return at the end of the function
$IssuanceTransformRuleManualSP = @{}
.
[manual releases]
.
$IssuanceTransformRuleManualSP
```

If you want to add a manual release we recommend you add the following code (where the manual releases is above):

```
### Description of the SP
        $TransformRules = [Ordered]@{}
        $TransformRules.[TransformRule Object] = $AllTransformRules.[TransformRule Object]

        $IssuanceTransformRuleManualSP["[EntityID for the SP"] = $TransformRules
###
```

To find which TransformRule Objects that are available run the following command:

```
PowerShell

Get-ADFSTkTransformRuleObjects
```

This will list all available TransformRule Objects.

To see how the TransformRule Objects are build up, look at the Import-ADFSTkAllTransformRules.ps1 in the /private folder of the module.

### Known SP's that need fixes

#### General

There are some SP's that we know needs attention to be able to work. Before you add any of them, please make sure they don't work as-is.

The PowerShell code provided for each SP should be copied to the get-ADFSTkManualSPSettings.ps1 script in the /config folder.  Please note the additional steps might also be needed.

#### Terena

```
    ### Terena.org/sp
        $TransformRules = [Ordered]@{}
        $TransformRules.'transient-id' = $AllTransformRules.'transient-id'
        $TransformRules.eduPersonTargetedID = $AllTransformRules.eduPersonTargetedID
        $TransformRules.eduPersonPrincipalName = $AllTransformRules.eduPersonPrincipalName
        $TransformRules.mail = $AllTransformRules.mail
        $TransformRules.displayName = $AllTransformRules.displayName
        $TransformRules.givenName = $AllTransformRules.givenName
        $TransformRules.sn = $AllTransformRules.sn
        $TransformRules.eduPersonScopedAffiliation = $AllTransformRules.eduPersonScopedAffiliation
        $IssuanceTransformRuleManualSP["https://terena.org/sp"] = $TransformRules
    ###
```

## Orcid

```
    ### orcid.org
        $TransformRules = [Ordered]@{}
        $TransformRules.eduPersonUniqueID = $AllTransformRules.eduPersonUniqueID
        $IssuanceTransformRuleManualSP["https://orcid.org/saml2/sp/1"] = $TransformRules
    ###
```

## Sectigo (Cert-manager)

Sectigo needs eduPersonEntitlement = urn:mace:terena.org:tcs:personal-user for all AL2 users.

Below is an example where the AL2 is retrieved from an AD group. Change the code based on how AL2 is stored in your institution.

```
    ### Cert-manager (Sectigo)
        $TransformRules = [Ordered]@{}
        $TransformRules.eduPersonPrincipalName = $AllTransformRules.eduPersonPrincipalName
        $TransformRules.displayName = $AllTransformRules.displayName
        $TransformRules.givenName = $AllTransformRules.givenName
        $TransformRules.mail = $AllTransformRules.mail
        $TransformRules.sn = $AllTransformRules.sn
        $TransformRules.schacHomeOrganization = $AllTransformRules.schacHomeOrganization
        $TransformRules.eduPersonEntitlement = [PSCustomObject]@{
           Rule=@"
              @RuleName = "Set eduPersonEntitlement for AL2 users"
                            c:[Type == "http://schemas.xmlsoap.org/claims/Group", Value == "<group name containing
all AL2 users>"]
                            => issue(Type = "urn:oid:1.3.6.1.4.1.5923.1.1.1.7", Value = "urn:mace:terena.org:tcs:
personal-user", Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/attributename"] = "urn:
oasis:names:tc:SAML:2.0:attrname-format:uri");
                            "@
              Attribute="http://schemas.xmlsoap.org/claims/Group"
        }

        $IssuanceTransformRuleManualSP["https://cert-manager.com/shibboleth"] = $TransformRules
    ###
```

## InAcademia

```
    ### Inacademia
        $TransformRules = [Ordered]@{}
        $TransformRules.transientid = [PSCustomObject]@{
        Rule=@"
            @RuleName = "synthesize persistent-id"
                c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid"]
                => add(store = "_OpaqueIdStore", types = ("urn:adfstk:persistentid"), query = "{0};{1};{2}",
param = "ppid", param = c.Value, param = c.OriginalIssuer);
            @RuleName = "issue persistent-id"
                c:[Type == "urn:adfstk:persistentid"]
                => issue(Type = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier",
                Issuer = c.Issuer,
                OriginalIssuer = c.OriginalIssuer,
                Value = c.Value, ValueType = c.ValueType,
                Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/format"] = "urn:oasis:
names:tc:SAML:2.0:nameid-format:persistent",
                Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/spnamequalifier"] =
"[ReplaceWithSPNameQualifier]",
                Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/namequalifier"] =
"http://$($Settings.configuration.StaticValues.ADFSExternalDNS)");
         "@
         Attribute=""
         }
         $IssuanceTransformRuleManualSP["https://inacademia.org/metadata/inacademia-simple-validation.xml"] =
$TransformRules
    ###
```