

Bulk Import of users with rolemembership

Use Case request: Bulk load new users by importing a CSV file containing users with attributes and multiple roles into OpenIDM 4.0. It is assumed the roles already exist within OpenIDM.

Example of CSV format:

```
uid,username,FirstName,LastName,EMail,Status,roles
cgarcia,cgarcia,Carlos,Garcia,carlos.garcia@forgerock.com,active,AVS-API|employee|AVS2-API
```

The last field, the roles field contains multiple roles separated by "|" pipe character.

Many thanks to Jake Feasel for figuring out the hard part, the transformation script.

Step-by-step guide

This effort was based on OpenIDM4.0

1. Create a CSV File Connector configured against the CSV import file.

The screenshot shows the configuration page for a CSV File Connector. It has tabs for 'Details', 'Object Types', and 'Sync'. Under 'General Details', the 'Enabled' checkbox is checked and the 'Connector Type' is set to 'CSV File Connector - 1.5.0.0'. Under 'Base Connector Details', the 'CSV File' path is '/opt/w/...csv', 'Header UID' is 'uid', 'Header Password' is empty, 'Quote Character' is '"', 'Field Delimiter' is ',', and 'Newline String' is empty.

2. Create a new mapping from the CSV connector to the Managed User Object.

The screenshot shows the mapping configuration page. The source is 'system/BulkLoad/_ACCOUNT_' and the target is 'managed/user'. The 'Attributes Grid' shows the following mappings:

SOURCE	TARGET	Link Qualifier
uid	userName	x
EMail	mail	x
LastName	sn	x
FirstName	givenName	x
Status	accountStatus	x
roles	/roles	x

Buttons at the bottom include 'Undo All Changes' and 'Save Properties'.

- On the Attributes Grid map each CSV column to the managed user target attribute. The main work is around mapping the multi-values roles from the CSV into the managed roles object. A transformation script is required to split the multiple roles and create the role memberships on openIDM.

Target Property: /roles

Property List Transformation Script Conditional Updates Default Values

Enter a transformation script for complex properties.

Current value for "source": roles

Type

Inline Script

```

1 source.split(',')
2 .map(function (role) {
3   return {
4     _ref: 'managed/role/' + (
5       openidm.query('managed/role', {'_queryFilter': '/name eq "' + role + '"'}).result
6     )
7   }
8 }

```

File Path

Add passed variables

The script used is:

```

source.split(',')
.map(function (role) {
return {
_ref: 'managed/role/' + (
openidm.query('managed/role', {'_queryFilter': '/name eq "' + role + '"'}).result[0]._id
});
}}

```

- Update the Behavior Policies to Create when missing and run reconciliation. Upon completion, check Managed Users and new users should be created as well as roles membership added.

USER **cgarcia**

Details Password Provisioning Roles Authorization Roles Direct Reports Linked Systems

<input type="checkbox"/>	DETAILS	TYPE
<input type="checkbox"/>	AVS-API	Role
<input type="checkbox"/>	employee	Role
<input type="checkbox"/>	AVS2-API	Role

{ < > }



Related articles

- [Amazon RDS as an Identity Management Repository](#)
- [Developing Against DB2](#)
- [Change mail example for IDM 6.5](#)
- [Bulk Import of users with rolemembership](#)
- [OpenIDM Client_Cert Auth Module and Jetty wantClientAuth Setting Dependency](#)