# AMX RBAC and Role Management Tutorial

This tutorial is for IT staff who are experienced in identity management, it requires insight into how the Active Directory works, and a working knowledge of Windows.

This exercise will show some of the more advanced features of AMX, specifically:

- Active Directory roles. Changing roles and updating roles using templates or role models.
- Role change hysteresis, changing group membership using make before break.
- Role management modes, Strict, Loose and None.
- Manager roles.

AMX runs on Windows and must be setup as shown in the AMX Tutorial Setup document. In this tutorial identityReport and identitySync are run from the Command Line using AMXRun which sets the environment variables, in production it is expected to be run by the Task Scheduler.

## 1. Active Directory Roles

This tutorial shows how a person's identity attribute can be mapped to a role template in the ActiveDirectory to add and remove them from groups associated uniquely with the role.

1. This tutorial uses the identityReportAD1.csv created in tutorial AD1. If necessary, recreate it with identityReport as described in Tutorial AD1.

2. Select a role attribute and update the ActiveDirectory2.Properties file. This is the attribute name in the Metaverse. For example this can be:

a. A location = location
b. A job title = title
c. A department = department

Update for example

```
ActiveDirectoryRoleAttribute1-1 = title
```

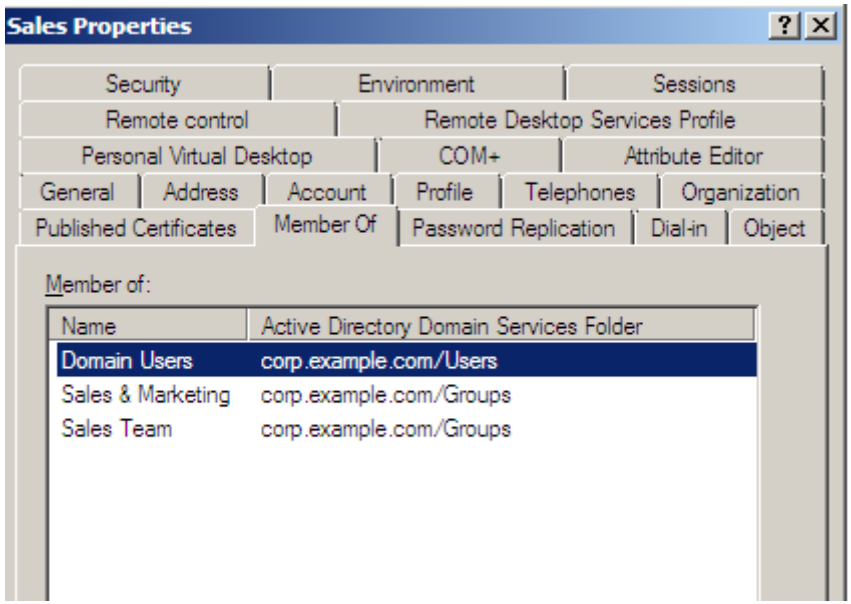
3. Update the ActiveDirectory2.properties with the location of the OU which will be created in step 3 to contain role templates

```
ActiveDirectoryRoleContainer1 = OU=roles,OU=AMX
```

The other parameter is RoleGroup which is the ActiveDirectory attribute memberOf that defines group membership

Optionally use ActiveDirectoryRoleAttribute1-2 = location etc to add more roles. Notice that it is not necessary to repeat `ActiveDirectoryRoleContainer1` when it is the same as the previous values.

4. Create the Roles OU in the active directory in the ActiveDirectory container defined in step 2. Add some template roles (users) in the OU with names matching the values found in the role attribute in step 1. For example:
Roles based on Title = Sales, Marketing, Operations, PMO, Finance
Template role location = London, Edinburgh, Leeds, Glasgow




5. Notice the use of the rolegroup property. This specifies the multivalued attribute in the Metaverse that will contain the groups associated with an individual's role and found in their ActiveDirectory object. It must match an attribute in the Active Directory schema. For example in the ActiveDirectory2.properties file, the property:

```
RoleGroup1-1 = memberOf
```

Corresponds to the attribute in the ActiveDirectorySchema1.txt

```
memberOf,memberOf;delta
```

6. Run identitySync.exe in the analyse mode. Check info.txt, review the roles that were found to see if they match those that are expected. Where there are roles that are not defined in the ActiveDirectory RoleContainer, info.txt might contain:

```
Warning: Active Directory GetRoleGroups. Cannot find Role hr admin 1 in OU=roles,OU=AMX
Warning: Active Directory GetRoleGroups. Cannot find Role hr admin 2 in OU=roles,OU=AMX
```

7. Check ActionFile.txt to see the accounts that will be added to groups.

### Updating a Role

This tutorial shows the effect of changing a role by adding a new group to the role template.

1. In the Active Directory, add a new group to an existing role that is already used by some accounts.

2. Run identitySync.exe in the analyse mode. Check ActionFile.txt to see the accounts that will be added to the new group.

### Strict, Loose and No group Management

Management of roles can be done in 3 ways, and defined as Types in the metaverse. This is intended to assist with the introduction of AMX into a production environment.

1. None, comment out ActiveDirectoryRoleAttribute1-1 in ActiveDirectory2.properties and no role management will be performed.

2. Loose, Users are immediately added to groups but never removed.

3. Strict, Users are immediately added to groups based on their roles, and they will be removed at some time in the future based on the value in days of RoleDeleteDelay. This is the normal production configuration.

   Strict roles have “strict” in their descriptions. Unless a role has “strict” in its description it defaults to Loose. All the groups defined in the role are then managed strictly unless they are also part of a Loose role. Loose takes precedent over Strict for each individual group.

The usual release to production process is to run identitySync in Loose mode and modify roles individually to Strict with a long RoleDeleteDelay. See AMX Role documentation for further details and approaches to creating Roles.

To show this:

1. Add “strict” to a role definition in the Active Directory.

2. Update ActiveDirectory2.properties, set RoleDeleteDelay to 7 (the default), and run identitySync.exe in the analyse mode, check ActionFile.txt and see that the memberAdd of the new group will be done today and the memberDel will be done a week later.

3. Remove “strict” from the role definition (Loose mode) and run identitySync.exe in the analyse mode, check ActionFile.txt and see that the memberDel has been removed.

### Hysteresis in Strict Roles

In production situations the timing for changing a person’s role and consequently access rights can cause disruption. When this is done too early the person may be unable to complete the tasks for the old role, and a role change is sometimes not a sudden

break from the old responsibilities. To avoid this sort of disruption AMX has role hysteresis which changes role in the following way:

1. The new groups are added immediately that the role changes. This allows the person to access documents and do some background reading concerning the new role.

2. For a defined period of time the person is a member of their existing groups and the new groups associated with the new role.

3. After the person starts the new role they are still able to access documents from their old role to answer occasional questions concerning it.

4. After the period of time has expired, the groups associated with the old role are removed. The length of time that the person has both roles is defined in the identitySync properties as RoleDeleteDelay and is measured in days.

To show this feature.

1. Update ActiveDirectory2.properties, set RoleDeleteDelay to 1 (day), this will remove groups from the old role tomorrow. A RoleDeleteDelay of 0 will add and remove the groups today.

2. Add a common group to 2 roles, for example Sales & Marketing to both the Sales and the Marketing group. Check that there are at least 1 person with the role Sales and another with Marketing. Add “strict” to the description of the Sales role.

3. Run identitySync.exe in the do mode to add the new group. Check that the users have been added to the Sales & Marketing group

4. Update identityReportAD1.csv, moving one of the persons from Sales to Marketing by updating the Title attribute when the roles are defined by Title as above.

5. Run identitySync.exe in the analyse mode. Check that one person will be added to the Marketing groups, and that they will be removed from the Sales group tomorrow. Also note that the person will remain a member of the Sales & Marketing group which is common between the two roles.

6. Run identitySync.exe in the do mode, note that the memberDel  for tomorrow remains.

7. Delete the ActionFile.txt file and run identitySync.exe in the analyse mode, note that the memberDel for tomorrow is recreated.

## Manager

After each of the sources of identities is loaded, they are evaluated to see if a person is identified as the manager in any other identity record. Where a person is identified as a manager, the attribute indicated by the IsaManager flag is set to “IsaManager”. This value can be replaced with the name of a valid role and the manager will be added to the groups defined by the role. This tutorial shows managers based in different locations being given roles that could add them to a local manager’s group and an organisation wide group.

1. Update IdCSVSchaemaAD1.txt, check that the ManagerJoin flag is correctly set, in this case to distinguishedName.

```
distinguishedName,distinguishedName;managerJoin
```

2. Add an evaluated Metaname such as isaManager with the flag IsaManager and the evaluation to replace “IsaManager” with the name of the role. For example “Manager” in this case:

```
,isaManager;IsaManager;replace/IsaManager/Manager/
```

3. Update ActiveDirectory2.properties.txt, add a new role Manager

```
RoleAttribute1-2 = isaManager
```

4. Create a Manager group in the Active Directory

5. Add a Manager role to the Active Directory as above and add it to the new group.

6. Run identitySync.exe in the analyse mode and review ActionFile.txt to see the users that would be added to the manager group.

### *No groups Added*

Check that the role was found. A warning message is printed on the console and in info.txt where a role does not match the value of the RoleAttribute.

```
Warning: Active Directory GetRoleGroups. Cannot find Role hr admin 1 in OU=roles,OU=AMX
```

Check that the role template is a member of a group and the user currently is not.

Using a log level of 2 or more search in debug for “GetManagedGroups” check that the group you expect to be managed has been identified.

```
ActiveDirectory GetManagedGroups CN=Managers,OU=Groups,DC=corp,DC=example,DC=com
```

```
ActiveDirectory GetManagedGroups CN=London Manager,OU=Groups,DC=corp,DC=example,DC=com
ActiveDirectory GetManagedGroups CN=Sales & Marketing,OU=Groups,DC=corp,DC=example,DC=com
ActiveDirectory GetManagedGroups CN=Marketing Edinburgh,OU=Groups,DC=corp,DC=example,DC=com
ActiveDirectory GetManagedGroups CN=Marketing London,OU=Groups,DC=corp,DC=example,DC=com
ActiveDirectory GetManagedGroups CN=Sales Team,OU=Groups,DC=corp,DC=example,DC=com
```

## Local Manager

This tutorial will show how a local manager role can be created. This could add the manager to a local manager group and the organisation wide group.

1. Update CSVSchema1.txt, modify the isaManager attribute. Prefix the attribute value with the value of the location or similar attribute using concat, and then replace the value returned by isaManager by manager. This will create roles like LondonManager:

   ```
   ,isaManager;IsaManager;concat:%location%;replace/IsaManager/Manager/
   ```

2. Create a London Manager group in the Active Directory

3. Assuming that the role will be LondonManager, create a Role Template with a matching name in the Active Directory and add it to the London Manager group.

4. Run identitySync.exe in the analyse mode and review ActionFile.txt to see the users that would be added to the London Manager group.