AMX

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 demonstrate some of the more advanced features of AMX and the Active Directory, specifically:

- Using a LDIF file from a managed resource, the Active Directory
- Using the LDAP adapter on an ADAM instance
- Multiple Active Directory instances

1. Setup

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.

2. Using LDIF

This exercise will use an LDIF extract from the Active Directory to demonstrate the capabilities of the LDIF adapter as a managed system. It will use the Identity.csv file extracted in tutorial AD1 as a source of identities.

1. Extract an LDIF file from the Active Directory of the users that were reported in Tutorial 1. For example users in OU=accounts.

```
ldifde -f example.ldf -p Subtree -r "(objectClass=User)" -d "ou=accounts,dc=corp,dc=example,dc=com"
      Connecting to "dc.corp.example.com"
      Logging in as current user using SSPI
      Exporting directory to file example.ldf
      Searching for entries...
```

```
Writing out entries.....................................................
.......................................
105 entries exported

The command has completed successfully

C:\Users\Administrator>
```

If a filter was used, this cannot be replicated in ldifde, so repeat the identityReport extract with the filter removed (blank).

2. Review LDIF1.properties, note the LDIF file.
   ```
   LDIFResource1 = example.ldf
   ```

3. Review LDIFSchema1.txt, notice that sAMAccountName has been used as accountName rather than uid and active is derived from userAccountControl rather than inetUserStatus. sAMAccountName and userAccountControl are both Active Directory specific LDAP schema extensions. This behaviour is controlled by LdapType1 = ADAM.

4. Run identitySync.exe in the analyse mode. Check there are no changes in the ActionFile.txt

3. LDAP

This exercise will use the Active Directory as a LDAP Server. The Active Directory has a non-standard LDAP schema, if an alternative LDAP server is available that can be used. Set LdapType to ADAM for the Active Directory, for other LDAP servers leave it blank. The manual LDIFfile uses the lowest common denominator syntax, which is that modify / delete, add or replace are used. Modify / replace will not add a missing attribute and set its value on all LDAP servers, notably ADAM. Modify / add has to be used and likewise modify / delete. See Redhat Directory Server 2.4.3.

1. Review LDAP1.properties, note the LDAP resource and modify to suit.

```
LdapResource1 = ldaps://dc1.example.com
LdapName1 = ADldap
LdapAccountContainer1 = ou=accounts,dc=corp,dc=example,dc=com
LdapSchema1 = LDIFSchema1.txt
LdapUser1 = cn=administrator,cn=users,dc=corp,dc=example,dc=com
LdapPasswd1 = LDAPpasswd1.txt
LdapDeletedContainer1 = OU=Deleted,OU=AMX,DC=corp,DC=example,DC=com
LdapType1 = ADAM
```

   Change other properties as necessary including the LdapUser1 as a full distinguished name and the account container. Create an LDAPPasswd1.txt file adding the password for LdapUser1 in the first line. The password will be encrypted when identitySync.exe is run. If this password is the same as ActiveDirectoryPasswd1.txt, it could be copied and renamed.

   Use the same schema as the LDIF resource.

   The LDAP server certificate must be trusted by the system running identitySync.

   If Tutorial AD1 has been completed and ADAM is being used, the source of identities IdentitiyReportAD1.csv can be used. Alternatively identityReport can be used to create a new identity file by:

   - Editing LDAP1.properties, commenting out the CSVidentityResource1, perhaps changing the Report property and running identityReport.

   - Uncomment CSVidentityResource1 and set it to use the file created above.

2. Run identitySync.exe in the analyse mode. Check there are no changes in the ActionFile.txt

```
C:\AMX\Tutorial1>identitySync.exe LDAP1.properties
Warning: Not run as administrator
Begins Mon, 02 May 2016 10:56:01 GMT  analyze
CSVidentity 1 C:\AMX\Tutorial1\IdentityReportAD1.csv
Last updated 02/05/2016 10:40:37
Extracted 103 Identities
Total of 103 Identities

LDAP 1 ldaps://dc.corp.example.com
Paged read
..
Extracted 102 Accounts
Account joins    102
Account creates  0
Account updates  0
Account disables 0
Account deletes  0
LDAP Manual Update LDIF do File LDIFdo.txt
LDAP Manual Update LDIF undo File LDIFundo.txt
LDAP Finished Mon, 02 May 2016 10:56:02 GMT
Ends Mon, 02 May 2016 10:56:02 GMT

C:\AMX\Tutorial1>
```

3. Make a change to the file referenced by CSVidentityResource1, for example IdentityReportAD1.csv

4. Run identitySync.exe in the analyse mode. Check the change is reflected in the ActionFile.txt. Check the LDIFdo.txt file also contains the change. If the change added a new user, notice the password is reported in the LDIFdo.txt file. This is a security risk after the LDIF file has been loaded into the server, and the file should be deleted after the passwords have

been distributed. If this is a security risk, blank LdapPasswordTemplate1 and no password will be created. Notice that when no password is set the new user remains disabled.

5. Load the LDIFdo.txt into the Active Directory.If the computer is a member of the domain it is not necessary to specify the server, it defaults to a domain controller. Notice that ldaps on port 636 is required when the password is set for a new user.

```
C:\AMX\Tutorial1>ldifde -i -f ldifdo.txt -t 636 -v -j \users\administrator
```

Bad credentials

```
Error: LDAP Extract The supplied credential is invalid. for ADldap
```

Either password or LdapUser1 is incorrect. LdapUser1must be in the form of a distinguished name

```
LdapUser1 = cn=administrator,cn=users,dc=corp,dc=example,dc=com
```

Failed to import any accounts

```
Error: LDAP Extract response = Referral
```

Attribute LdapResource1does not match the domain

Failed to Connect

Error: LDAP Extract. The LDAP server is unavailable for <host>:636

Try using ldap:// on port 389 for attribute LdapServer1. If this works:

- The server may not have a certificate installed. It needs a server authentication certificate.
- The server's certificate may not be trusted by the system running identitySync.
- Hostname must be fully qualified and resolvable with dns. Use nslookup to check.

Unwilling to perform

```
Add error on entry starting on line 17: Unwilling To Perform
The server side error is: 0x1f A device attached to the system is not functioning.
The extended server error is:
```

```
0000001F: SvcErr: DSID-031A120C, problem 5003 (WILL_NOT_PERFORM), data 0
```

Caused by trying to set a password when not using SSL on port 636.

Error: LDAP Export Create Exception The server cannot handle directory requests.

Caused at least by a password being in the wrong format. For example when the ldapType is incorrect.

Changing LdapPageSize

The Page ldap Control is used to read the directory in multiple pages, by default each page is 100 entries. This can be varied to improve the performance of the LDAP extract. The upper limit depends on the LDAP implementation, typically it is 5,000 – the Administrative Limit, which can be exceeded by an administrator account. Larger values increase the load on the LDAP directory, smaller values increase the load on the client.

Changing LdapSearchFilter

The LdapSearchFilter is the RFC implementation. Default is (objectClass=Person), Microsoft AD and ADAM are more efficient if (objectCategory=Person) is used because the attribute is indexed.

Note that unlike other LDAP implementations, neither Microsoft AD nor ADAM can use the extension (&(objectCategory=Person)(|(ou:dn:=LON)(ou:dn:=EDI).

4. Multiple Active Directory Instances

In tutorial AD1.2 the ActiveDirectoryFilterValue1 was set to extract accounts from 2 OUs. This tutorial shows 2 separate extracts, one for each OU. This would be used when the OUs were managed by different parts of the organisation who used different account management standards, needing an ActiveDirectorySchema for each OU. For example if the account description attribute in one OU included the department, but in the other the department attribute was used. Instance ActiveDirectoryResource2 uses ActiveDirectorySchema2.txt. Notice also in the Active Directory properties file that the ActiveDirectoryAccountcontainer is the

same for both instances. This has an effect on the unique attribute modifier. Unique adds all records to the unique table including records that are filtered out, in this case the schema of the second instance cannot have the Unique attribute modifier otherwise adding the unique values to the table will fail the uniqueness test the second time because the value was added by the first instance.

In the example above the Identity schema will have to use the department attribute to create two metaverse attributes, one for each format. Each ActiveDirectory instance will then use its schema to map its metaverse attribute to its ActiveDirectory attribute. In situations where attributes are differently formatted and there is agreement to standardise them, identitySync could be used to update the non-standard attributes of one of the instances.

In situations where the forest has multiple domains, for example EMEA and AsiaPac these should be managed independently by two versions of identitySync property files. This version of AMX is not able to move accounts between domains, though code has been developed and used in a Microsoft FIM R2 implementation and could be incorporated into identitySync.

1. Update identitySync Properties

Edit the ActiveDirectory2.properties file, remove ou=lon from:

```
ActiveDirectoryFilterValue1 = ou=EDN:ou=lon
```

Uncomment the second instance of Active Directory, adding:

```
ActiveDirectoryExtractMode1 = Add
ActiveDirectoryResource2 = dc1.example.com
ActiveDirectoryName2 = corp
ActiveDirectoryFilterValue2 = ou=LON
```

Set the ExtractMode to Add for the first instance.

```
ActiveDirectoryExtractMode1 = Add
```

All the other parameters for the second instance remain the same, and by default will use the values defined for the first instance (including ExtractMode, FilterAttribute and AccountContainer).

The ExtractMode “Add” adds the accounts from the second instance to the first. ExtractMode “Single” is the default and causes each instance of the ActiveDirectory to be analysed individually.

2. Run identitySync in the analyse mode and check that there are no additional changes.

```
C:\AMX\Tutorial1>identitySync.exe ActiveDirectory2.properties
Warning: Not run as administrator
Begins Mon, 02 May 2016 11:14:22 GMT  analyze
CSVidentity 1 C:\AMX\Tutorial1\IdentityReportAD1.csv
Last updated 10/09/2014 16:51:36
Extracted 102 Identities
Total of 102 Identities

ActiveDirectory 1 dc.corp.example.com
Extracted 52 Accounts
ActiveDirectory 2 dc.corp.example.com
Extracted 50 Accounts
Account joins    102
Account creates  0
Account updates  0
Account disables 0
Account deletes  0
ActiveDirectory Finished Mon, 02 May 2016 11:14:24 GMT
Ends Mon, 02 May 2016 11:14:24 GMT

C:\AMX\Tutorial1>
```

3. Update Second Active Directory Instance. Make a change to an attribute of one person with an account in the second instance of the Active Directory in IdentityReportAD1.csv.

4. Run identitySync.exe in the do mode and check that the update is made. Run identitySync.exe in the undo mode and remove the change from IdentityReportAD1.csv.