AMX Account Management Tutorial CSV1

This tutorial shows how to use attribute modifiers to transform the values of any attribute. This is often required because attribute values in an authoritative source of identities do not necessarily match the values in a managed resource. For example the full names may need to be constructed from the first and last names and they may be formatted differently.

In this tutorial identityReport and CSV schemas will be used with a simple CSV file of identities to show some of the features of attribute value transformation in AMX. The file of identities has been artificially created and does not represent the information usually received from an authoritative source. It consists of the left half representing identity information which is used by the identity schemas and the right half which represents a managed resource information. The transforms make the identity values match the account values

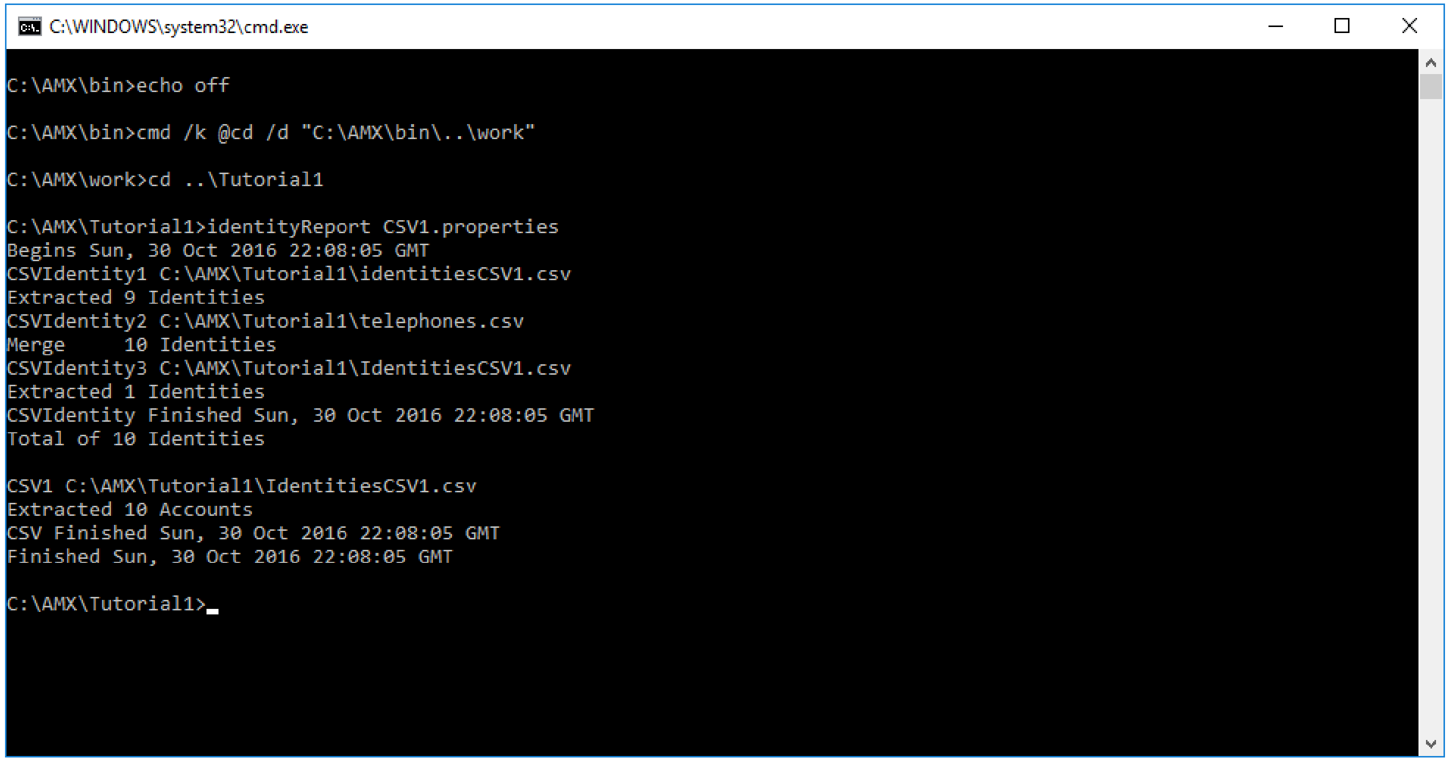
The tutorial will use configuration files that demonstrate:

1. Concatenating and changing case of attribute values
2. Using the concatIfNull attribute modifier to update blank attribute values
3. Using lookup tables
4. Appending value that may be null
5. Creating distinguishedNames with concat
6. Creating account names with truncate
7. Using an EndDate to mark an identity as inactive
8. Merging attributes such as telephone numbers
9. Reformatting attribute values such as telephone numbers
10. Setting the IsaManager attribute

11. Setting the manager's email of an identity

The configuration will then be used by identityReport to create a report CSVReport.csv that shows the transform of the identity attributes. identitySync can also use the same configuration files to show the differences in ActionFile.txt.

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.



```
C:\AMX\bin>echo off

C:\AMX\bin>cmd /k @cd /d "C:\AMX\bin\..\work"

C:\AMX\work>cd ..\Tutorial1

C:\AMX\Tutorial1>identityReport CSV1.properties
Begins Sun, 30 Oct 2016 22:08:05 GMT
CSVIdentity1 C:\AMX\Tutorial1\identitiesCSV1.csv
Extracted 9 Identities
CSVIdentity2 C:\AMX\Tutorial1\telephones.csv
Merge     10 Identities
CSVIdentity3 C:\AMX\Tutorial1\IdentitiesCSV1.csv
Extracted 1 Identities
CSVIdentity Finished Sun, 30 Oct 2016 22:08:05 GMT
Total of 10 Identities

CSV1 C:\AMX\Tutorial1\IdentitiesCSV1.csv
Extracted 10 Accounts
CSV Finished Sun, 30 Oct 2016 22:08:05 GMT
Finished Sun, 30 Oct 2016 22:08:05 GMT

C:\AMX\Tutorial1>_
```

The demo uses a file of identities and accounts in identitiesCSV1.csv. It is best viewed with Excel. The first leftmost columns are identities and the attributes are transformed in this tutorial to match the account attributes in the right most columns. The tutorial showcases attribute transform and does not represent a production situation. Usually identities are extracted from an authoritative identity source, transformed and then used to load managed resources.

The attribute transformation is defined in the schema files, specifically IdCSVSchemaCSV1 and IdCSVSchemaCSV2. These are used to extract the data from files identitiesCSV1.csv and telephones.csv and build a database of identities in the metaverse. The schemas define the staging attribute name, the metaverse attribute name and the transform of each attribute. identityReport uses CSV1.properties to create a report which is sorted by EmployeeID and written to a report file IdentityReportCSV1.csv.

The attribute transforms are fully described in the reference document. Understanding how they operate is best done by changing them and checking the identityReport report file. The debug.txt file also has some insight of the transform. As configured the transformed identities match the accounts so if identitySync is run there will be no changes written to the ActionFile.txt file, which is used during the load phase to make the changes to a managed resource.

1. Concatenating and changing case of attribute values

Open IdCSVSchemaCSV1.txt. This shows the creation of email addresses by first creating new metaverse attributes first_lower and last_lower which are used to create a third new metaverse attribute called mail.

```
firstName,first_lower;toLower
lastName,last_lower;toLower
,mail;concat:%first_lower%.%last_lower%@example.com
```

2. Concatenating if Null to update blank attribute values

In situations where the managed resource uses preferred names where these are available the ConcatIfNull attribute flag can be used. This only updates a blank value, and won't overwrite an existing one. In the identities file Alban Wilson has a preferred name Al and this is used to create the distinguishedName.

```
preferredName,;ConcatIfNull:%firstName%
```

A more complex example is using a regular expression to extract an employeeID from a description and using it if the empNum attribute is blank. This is particularly useful in managed resources. AMX implements the .Net regex method and so all the power of regular expressions is available. See http://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx for a quick reference guide.

```
description,employeeID2;replace/[a-zA-Z()]+-*(\d*)/$1/;trim
empNum,employeeID;unique;managerjoin;ConcatIfNull:%employeeID2%
```

In this case the description attribute of A Wilson contains "Employee Number (204)", the replace matches one or more alphabetic characters and brackets with a decimal number surrounded by brackets and returns the decimal number in the Metaverse Attribute employeeID2 only replaces a blank value in employeeID. So for example if the blank value in empnum in identitiesCSV1.csv was changed to 2040 the employeeID2 value would not over-write it.

3. Using lookup tables

Lookup tables replace the value of a metaverse attribute with the corresponding value in the lookup table as found in city.csv. This value is used for the ou attribute, which in turn is used to build a distinguishedName in step 5.

```
location,ou;right0 ;lookup=city.csv
```

the attribute values in location are London or Edinburgh, these are looked up in city.csv which contains:

```
London,LON
Edinburgh,EDN
*,LON
```

When matches are found the value on the right side is returned. The default is marked with the *, in this case LON. Note that the identity A Wilson has a blank location attribute value, and in the report CSVreport.csv the default LON is used.

4. Appending value that may be null

In situations where for example an initial will be concatenated to a value, and the initial may be blank. Regular expressions can be used to add a space only when the initial has a value.

```
initials,initials+space;replace/[A-Z]/ $&/
```

The initial is used to create the distinguishedName in the next step.

5. Creating distinguishedNames with concat

The distinguishedName concatenates the last name and preferredName and the initial if there is one. It uses the metaverse attribute ou value which has derived in step 3.

```
,distinguishedName;concat:CN=%lastName%\,
%preferredName%%initials+space%,OU=%ou%,OU=accounts,DC=corp,DC=example,DC=com;unique
```

6. Creating account names with truncate

This shows how to use attribute modifiers to create initials and concatenate them with the last name to create an account name. This exercise is not realistic in a production situation, instead the unique account name creation process should be used, this is intended as a demonstration of attribute value manipulation. New metaverse attributes firstNameShort and lastNameSort are created using the truncate and then their case is changed to lower and apostrophes are removed. These are then concatenated to create a new metaverse attribute accountName.

```
firstName,firstNameShort;truncate2;toLower
lastName,lastNameShort;replace/'//;truncate10;toLower
,accountName;concat:%lastNameShort%%firstNameShort%
```

7. Using an EndDate to mark an identity as inactive

When an attribute is marked as an endDate string, the date is treated as an end date. If the date is in the future the value returned is “Y”, otherwise “N”.

```
endDate,active;endDate;Lookup:Active.txt
```

The attribute value is interpreted as an end date or last day of work. If the end date is yesterday the account is inactive, else if the date is today this is the last day that the account is active. The attribute value must be the native value of the end date such as 01/04/2016 on the system that is running AMX. If the end date does not specify the time this defaults to midnight. If the date is 01/04/2016 17:30:00 the identity will be disabled after 5.30pm when identitySync next runs.

If the disable time is critical and the default midnight time is unsuitable use regex to replace the end of a date with a time, such as:

```
replace/(\S {10})/$1 17:30:00/
```

Regex will append 17:30:00 to a string of 10 characters, leaving blank values unchanged. Note that the concatenation attribute modifier cannot be used because it will always append the time, including end dates that are blank. Also concat is performed after end date is evaluated. The order of transform operations are:

After each attribute extract:

1. Transform

After each record extract

2. Concatenation
3. Filter

After all identity extracts

4. Merge identity extracts
5. Update IsaManager

See the reference manual for more information about the order of the transform operations.

8. Filtering Records

The properties file CSV1.properties has an example of filtering records. The file Identies1.csv is used first to extract records with employeeIDs of 200 to 240 and then a second time to extract records with employeeIDs less than 200. In practice the second and subsequent sources of identities would be separate files or sources, this is a simplistic example.

```
CSVIdentityResource1 = identitiesCSV1.csv
CSVIdentityname1 = Ids
CSVIdentitydelim1 = ,
CSVIdentityschema1 = IdCSVschemaCSV1.txt
CSVIdentityfilterAttribute1 = employeeID
CSVIdentityfilterValue1 = 200-240

CSVIdentityResource3 = identitiesCSV1.csv
CSVIdentityname3 = Id3
```

```
CSVIdentitySchema3 = IdCSVschemaCSV1.txt
CSVIdentityExtractMode3 = Add
CSVIdentityfilterAttribute3 = employeeID
CSVIdentityfilterValue3 = !200
```

The first extract gets records with employeeIDs having values between 200 and 240, the third gets records with employeeIDs less than 200. The third extract is an add so the net result is all identity records in identitiesCSV1.csv are reported. Any attribute can be used to filter records, for example the attribute OU could be used to filter records only from London. The order of transform operations is also significant when filtering, for example filtering by IsaManager is not possible because at the time of the filtering IsaManager has not been evaluated.

9. Merging attributes

The second extract is a merge, values found in this extract are merged into the existing identity extracts.

```
CSVIdentityResource2 = telephones.csv
CSVIdentitySchema2 = IdCSVschemaCSV2.txt
CSVIdentityExtractMode2 = Merge
CSVIdentityfilterAttribute2 =
```

The schema file must have a join flag

```
displayName,;join;displayName
telephoneNumber,telephone
```

The merge will update the telephone attribute of every identity that has the same displayName in both telephones.csv and the identity extracts. Merge is similar to lookup in step 3. It is different and more powerful when used to merge other resources, for example the Active Directory rather than a CSV file. The merge of an Active Directory is useful when the accountName is used to create accounts on database, ldap or other managed resources.

10. Reformatting attribute values such as telephone numbers

Regular Expressions can be used to reformat strings such as telephone numbers. See IdCSVSchemaCSV2.txt. This expression finds the first 4 digits, then the next 3 and finally anything from 3 to 6 digits. The finds are then rewritten with gaps between them.

```
telephoneNumber,telephone;replace/ //;replace/(\d{4})(\d{3})(\d{3,6})/$1 $2 $3/
```

Other formats such as ($1) $2-$3 could be used as required. See http://msdn.microsoft.com/en-us/library/az24scfc(v=vs.110).aspx for a quick reference guide to regular expressions.

11. Setting the IsaManager attribute

An identity's manager attributes are set after all the identities and accounts have been extracted if the IsaManager attribute flag is used to mark a derived metaverse attribute. The attribute is set to "isaManager" if the identity is found to be the manager of one or more other identities. The lookup changes this to True or False. Flags managerID and managerJoin must also be defined so that the metaverse can be searched for the empnum of an identity's manager in the attribute flagged by managerJoin.

```
manager,managerEmpnum;managerID
empNum,employeeID;managerJoin;unique
,isaManager;isaManager;lookup=ManagerStatus.csv
```

12. Setting the manager's email of an identity

Manager processing can set the manager's name and email address of an identity. In this instance the email is set. The managerID and managerJoin is also used to search for an identity's manager, and the mail attribute of the manager's identity record is copied into identity's attribute flagged with managerMail.

```
,mail;concat:%first_lower%.%last_lower%@example.com;Mail
,manager;managerMail
```

The corresponding flags for the name of an identity's manager are Name and ManagerName. ManagerName is used to update the manager attribute of an managed resource, for example the Active Directory uses distinguishedName.

Running IdentitySync

The identityReport report CSVReport.csv when run with the schemas supplied will show both identity and account record for each person are identitical.

```
Id3,103,Y,reidcl,Clyde Reid,"CN=Reid\, Clyde
A,OU=EDN,OU=accounts,DC=corp,DC=example,DC=com",clyde.reid@example.com,0131 257
7770,TRUE,calum.scott@example.com
CSV,103,Y,reidcl,Clyde Reid,"CN=Reid\, Clyde
A,OU=EDN,OU=accounts,DC=corp,DC=example,DC=com",clyde.reid@example.com,0131 257
7770,TRUE,calum.scott@example.com
```

Consequently when identitySync uses the same schemas and data there will be no changes.

```
C:\Dev\AMX\Tutorial1>identitySync.exe CSV1.properties
Warning: Not run as administrator
Begins Tue, 26 April 2016 10:36:56 GMT  analyze
CSVIdentity 1 C:\AMX\Tutorial1\identitiesCSV1.csv
Last updated 20/03/2016 10:10:28
Extracted 9 Identities
CSVIdentity2 C:\AMX\Tutorial1\telephones.csv
Last updated 20/03/2016 20:56:55
Merge     11 Identities
CSVIdentity3 C:\AMX\Tutorial1\IdentitiesCSV1.csv
Last updated 20/03/2016 10:10:28
Extracted 2 Identities
Total of 11 Identities

CSV1 C:\AMX\Tutorial1\IdentitiesCSV1.csv
```

```
Last updated 20/03/2016 10:10:28
Extracted 11 Accounts
Account joins    11
Account creates  0
Account updates  0
Account disables 0
Account deletes  0
Ends Tue, 26 April 2016 10:36:56 GMT
```