

AMX Account Management Tutorial CSV2

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, specifically:

- Using Excel as an Identity source
- Using a CSV file from a managed resource
- Using a SAP Text File as an additional Identity source
- Using identityReport to review attribute transformations
- Auditing changes made by identitySync
- Debugging

1. Setup

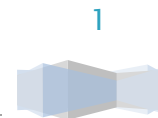
AMX must be setup as shown in Tutorial Setup. In this tutorial identityReport and identitySync are run from the Command Line using AMXRun which sets the environment variables. In this tutorial identityReport and identitySync are run from the Command Line using AMXRun which sets the environment variables.

Excel

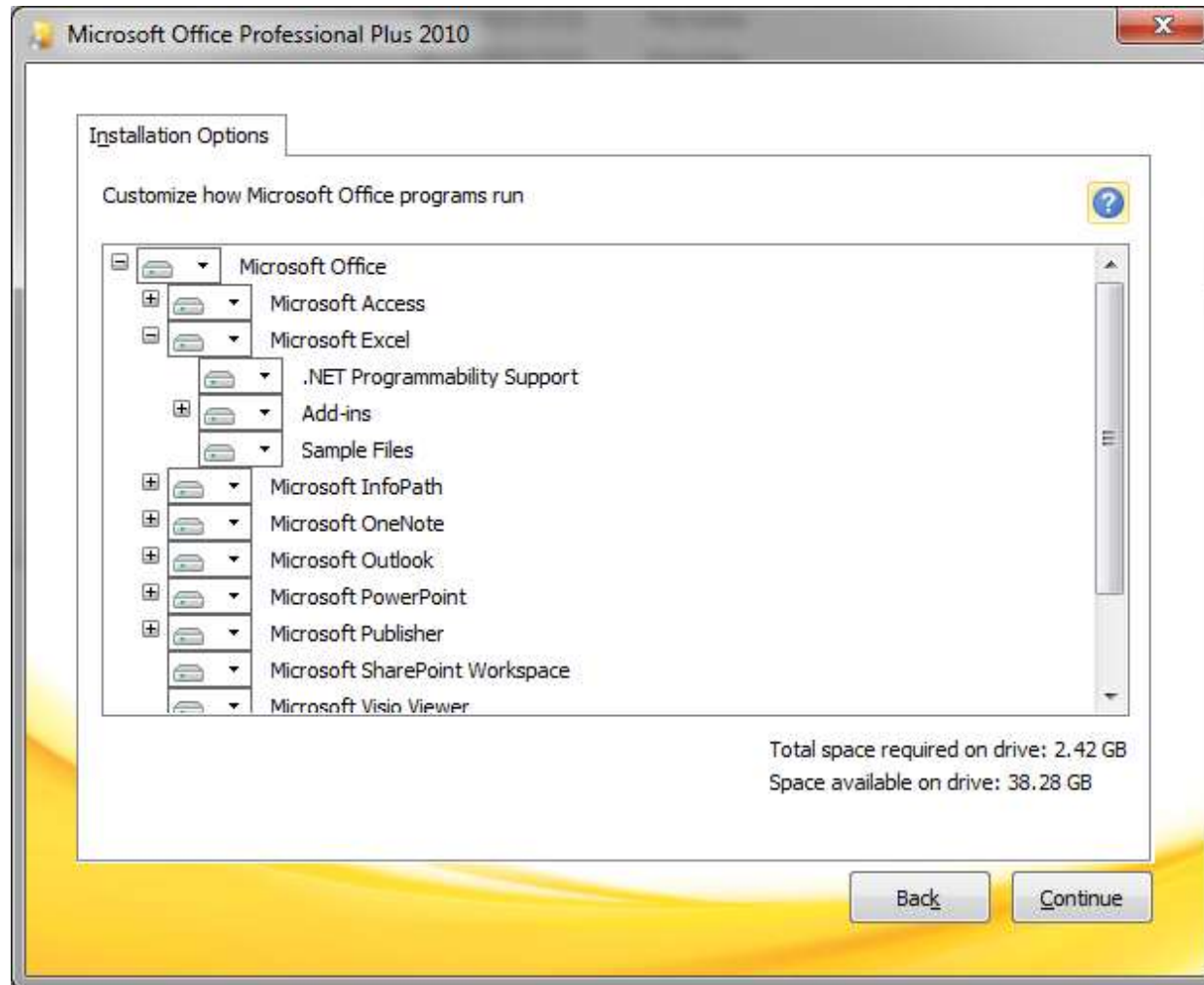
To run the Excel tutorial the Microsoft Office Interop Assemblies must be available.

The Assemblies are installed by default by Microsoft Office 2013.

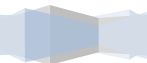
AMX includes the Microsoft Office Primary Interop Assemblies (PIA) Redistributable.



Alternatively install from the Microsoft Office distribution media Office 2010 .NET Programmability Support for the Excel.



Office Tools and Shared Libraries are not required. This will install the interop dlls.



2. Using Excel

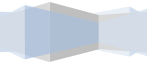
This exercise will show how Excel can be used as a source of identities.

1. This is a continuation of the previous tutorial CSV1, convert IdentityReport1.csv into IdentityReportCSV1.xlsx using Excel. Quit Excel.
2. Review Excell.properties note that 3 instances of Excel has been added using the same schema file as was used for the CSV version. No changes to the schema are required because the column headers are the same. The Excel adapters use the same filter as the CSV versions which are commented out.

```
ExcelResource1 = IdentitiesCSV1.xlsx  
Excelname1 = Excell  
Excelschema1 = IdCSVschemaCSV1.txt
```

3. Run identitySync.exe in the analyse mode, check that there are no unexpected changes caused by changing a CSV file for an Excel version.

```
C:\AMX\Tutorial1>identitySync.exe Excell.properties  
Warning: Not run as administrator  
Begins Mon, 02 May 2016 17:28:09 GMT analyze  
Excel 1 C:\AMX\Tutorial1\IdentitiesCSV1.xlsx  
Last updated 28/04/2016 14:52:16  
Extracted 9 Identities  
Excel 2 C:\AMX\Tutorial1\telephones.xlsx  
Last updated 28/04/2016 11:45:36  
Merge 11 Identities  
Excel 3 C:\AMX\Tutorial1\IdentitiesCSV1.xlsx  
Last updated 28/04/2016 14:52:16
```



```
Extracted 2 Identities
Total of 11 Identities
```

```
CSV1 C:\AMX\Tutorial1\identitiesCSV1.csv
Last updated 28/04/2016 14:57:28
Extracted 11 Accounts
Account joins      11
Account creates   0
Account updates   0
Account disables  0
Account deletes   0
Ends Mon, 02 May 2016 17:28:14 GMT
```

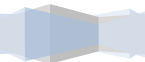
```
C:\AMX\Tutorial1>
```

4. Note that the Excel file can be encrypted with a password. Set the password in the first line of ExcelPasswd1.txt and it will be used to read the Excel file and then encrypted when identitySync.exe runs.

3. Using a SAP Text File as an additional Identity source

This shows the use of a short SAP Text file containing 3 identities added to identities in a CSV file, which are then synchronised with a CSV file. It is a continuation of the exercise above. The format of the SAP Textfile is attributeName : attributeValue, refer to SAP documentation for further details.

1. Open SAPI.txt to review the attributes. Attributes that can be used are:
 - a. firstName
 - b. lastName
 - c. preferredName
 - d. employeeID
 - e. title



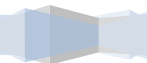
- f. department
- g. location
- h. manager
- i. telephoneNumber
- j. active, notice that in SAP the status is Active or Withdrawn. The TextFile adapter recognises these values.

2. TextFileSchema1.txt is used to extract the SAP.txt. Note that as in Tutorial CSV1 the distinguished name is derived from the identities names and location. Also:

- The manager is identified by their first and last name rather than their employee number as in the CSV file of identities. This is a common situation in a production environment.
- The Display Name attribute format is of the form lastName, firstName which is different to the format used in the CSV1 resource. The attribute modifier NoComma is used to reformat the value.

3. Run identitySync.exe in analyse mode and check for any changes, there should be none.

```
C:\AMX\Tutorial1>identitySync.exe TextFile1.properties
Warning: Not run as administrator
Begins Mon, 02 May 2016 17:33:13 GMT analyze
CSVIdentity 1 C:\AMX\Tutorial1\identitiesCSV1.csv
Last updated 28/04/2016 14:57:28
Extracted 8 Identities
CSVIdentity 2 C:\AMX\Tutorial1\telephones.csv
Last updated 28/04/2016 11:42:14
Merge      11 Identities
TextFile 1 C:\AMX\Tutorial1\SAP1.txt
Last updated 28/04/2016 16:10:28
Extracted 3 Identities
Total of 11 Identities
```



```
CSV1 C:\AMX\Tutorial1\identitiesCSV1.csv
Last updated 28/04/2016 14:57:28
Extracted 11 Accounts
Account joins      11
Account creates    0
Account updates    0
Account disables   0
Account deletes    0
Ends Mon, 02 May 2016 17:33:13 GMT
```

Multiple Textfiles can be used as sources of Identities.

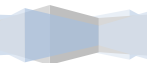
4. Cut one of the identities from SAP1.txt and add it to SAP2.txt
5. Update identitySync.properties to add
`TextFileResource2 = SAP2.txt`
6. Run identitySync.exe in analyse mode and review ActionFile.txt for any changes. There should be none.

4. Using identityReport to review attribute transformations

identityReport can be used to review the differences between the multiple sources of identities and the different transforms that they use. Run `identityReport TextFile1.properties` and open the report in `IdentityReportTextFile1.csv`

Note that it is sorted by the join attribute, in this case defined in `CSVSchemaCSV1.txt` as the `employeeId`.

Note that the first records are from the CSV identity source and the last 3 are from the TextFile source. They have been transformed to match each other.

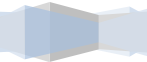


5. Auditing

AMX is designed to be run as a scheduled task, communicating with administrators using email. The info message format is:

```
Begins Mon, 02 Dec 2013 08:28:30 GMT do
CSVIdentity C:\AMX\Tutorial\IdentityReport1.csv
Last updated 10/09/2013 16:51:36
Extracted 102 Identities
Total of 102 Identities

ActiveDirectory1 corp1
Extracted 50 Accounts
ActiveDirectory2 corp2
Extracted 52 Accounts
Account joins      101
Account creates    1
Account updates    7
Account disables   2
Account deletes    1
ActiveDirectory #1 corp1
ActiveDirectory Load corp1 Delete JonesB
ActiveDirectory Load corp1 Update do WilsonA => description=Test;location=;title=Marketing
ActiveDirectory Load corp1 Update do ThomsonA => description=Test;title=Sales
ActiveDirectory Load corp1 Update do engela => distinguishedName=cn=anette
engel,ou=edn,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory #2 corp2
ActiveDirectory Load corp2 Update do CampbellA => distinguishedName=cn=angus
campbell,ou=edn,ou=accounts,dc=corp,dc=example,dc=com;location=Edinburgh
ActiveDirectory Load corp2 Update do WrightA => manager=cn=bonnie
burns,ou=edn,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory Load corp2 Create cn=brian jones,ou=lon,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory Load corp2 Update do SmallA => active=N
ActiveDirectory Load corp2 Update do SmallA1 => active=N
Ends Mon, 02 Dec 2013 08:28:31 GMT
```



These info messages are expected to be checked by an administrator and then moved to a mail folder where they can be searched during an audit.

AMX also writes an audit file when identitySync.exe is run in the do or undo mode. The audit file is not tamperproof (unlike the Transaction File ActionFile.txt).

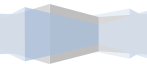
To create an Audit file:

1. Update ActiveDirectory2.properties, and set the name of the audit file, for example

```
LoggingAuditFile = C:\data\AMX\Audit2013\ActionAudit.txt
```

2. Run identitySync.exe in the do mode to write an entry in the audit file. Check the audit file, it will only contain actions that were successfully completed. The info file shows actions and their outcomes. For example:

```
02/12/2013 08:28:30
ActiveDirectory Load corp2 Delete JonesB
ActiveDirectory Load corp2 Update do WilsonA => description=Test;location=;title=Marketing
ActiveDirectory Load corp2 Update do ThomsonA => description=Test;title=Sales
ActiveDirectory Load corp2 Update do engela => distinguishedName=cn=anette
engel,ou=lon,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory Load corp2 Update do CampbellA => distinguishedName=cn=angus
campbell,ou=edn,ou=accounts,dc=corp,dc=example,dc=com;location=Edinburgh
ActiveDirectory Load corp2 Update do WrightA => manager=cn=bonnie
burns,ou=edn,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory Load corp2 Create cn=brian jones,ou=lon,ou=accounts,dc=corp,dc=example,dc=com
ActiveDirectory Load corp2 Update do SmallA => active=N
ActiveDirectory Load corp2 Update do SmallA1 => active=N
```



6. Debugging

The debug file as defined in `ActiveDirectory2.properties` has two features, the level of detail and the account name of one user to display. The debug file is mailed to the `DebugRecipients` when the `mailMode` is `< 2`, and a file called `debug.txt` is written to the current directory. For example

```
DebugSubject = Example Domain Debug
DebugRecipients =
LoggingLevel = 2
DebugUser = Small, Aileen
```

The `DebugUser` is reported when the `LogLevel` is 2 or greater.

LogLevels are:

- 1 debug log matches info. Minimal logging.
- 2 configurations, `debugUser`
- 3 details of configurations
- 4 identity and resource record names
- 5 details of identity and record attributes

LogLevel 5 includes a record of every identity and account extracted and therefore can be very large, we recommend that a test OU and a reduced identity set is used during extended debugging sessions using `LogLevel 5`.

The `debug.txt` is overwritten on every run.

