AMX accountMatch Reference

AccountMatch is intended to be used during the installation and configuration phase of an Identity Management project in organisations that cannot match all accounts to their owners. AccountMatch will recursively match accounts to persons or identities, it uses a file of identities and a file of accounts. Typically the file of identities is created by HR. AMX tools identitySurvey and accountSurvey are available to create them. The match rule uses combinations of attributes common in both files, for example first & last names, employee numbers, phone numbers and locations. AccountMatch is intended to be used during the release to production process to identify the owners of all the active accounts.

Features:

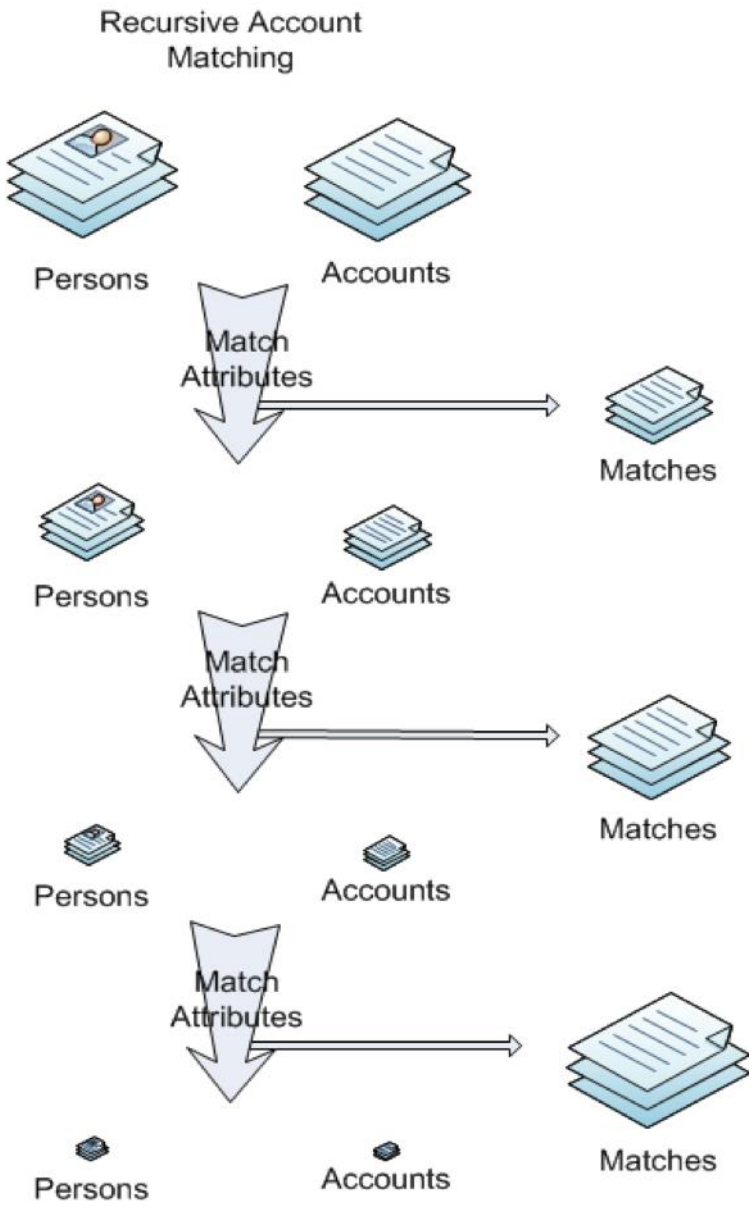
- Joins only unique matches, if more than one identity matches a single account or vice versa there is no match.
- It is most successful when there is a rich set of attributes to match, for example an HR report and the Active Directory.
- Has extensive facilities common to all AMX programs to manipulate attribute values before matching, including:
    - Splitting full names into first and last names, for example “Harold B Brown” split into “Harold ”,”Brown”.
    - Merging first and last names to create a full name. For example "Brown, Harold".
    - Using the preferred name in an HR report if there is one, in preference to the first name.
    - Looking up preferred names and their first name equivalents, for example "Harry" equivalent to "Harold".
    - Converting a middle name to an initial.
    - Reformatting attribute values, for example an account with a description "Staff-0075151" reformatted to "0075151" to match an employeeID.
    - If a consistent naming convention is in use, re-creating account names from first and last names in the identity source. For example when the account names are made from the first character of the first name and 7 characters from the last, Dave Robinson is matched with an account "drobinso".
- Uses in memory databases, extremely fast to run.
- Creates a match file using a formatting template. The match file could be a file of identity attributes and their matching account UIDs, which can be used in an Identity Management system, including of course identitySync. For example:

```
0075151,brownh
```

Or the template can be used to create manual update records such as an Active Directory dsmod update, that can be used to update or tattoo an account with its corresponding person identifier. For example :

```
dsquery brownh | dsmod –empID 0075151
```

- Recursively matches, using the results of a match as input to the next cycle. For example finding unique matches of first and last name, followed by preferred and last name, followed by last name and location, followed by matching first name to last and last to first (simple to do and in practice has found some matches in a multinational organisation). Finally when the volume is small enough a visual match. At each stage the results can be reviewed and checked.


Recursive Account Matching
Persons
Accounts
Match Attributes
Matches
Persons
Accounts
Match Attributes
Matches
Persons
Accounts
Match Attributes
Matches
Persons
Accounts

Runtime Arguments

accountMatch is run from a command prompt:

```
accountMatch.exe propertiesFile
```

Configuration

The properties file contains generic, application specific, and adapter specific properties.

Properties

1. AccountAttributes a comma separated list of attributes to write to AccountOutFile. Usually this can be copied from the header of the account CSV file. All the attributes listed in this property must be also in the schema as Meta Attributes.

2. AccountReport the name of the file that accountMatch will write un-matched accounts for subsequent runs.

3. AccountDelimiter the delimiter used in AccountOutFile, default is ",".

4. DelimList, the deliminator in Attribute values for lists. The default is "|".

5. identityAttributes a comma separated list of attributes to write to identityOutFile.

6. identityReport the name of the file that accountMatch will write un-matched identities for subsequent runs.

7. identityDelimiter the delimiter used in identityOutFile, default is ",".

8. identityUnique the attribute in the schema used to substitute %identityUnique% in the matchTemplate

9. LoggingLevel, 1 – 5, 1 = summary, 5 maximum. The logging methodology is:

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

10. LoggingType, LogType 1 write to info and debug log files only, LogType 2 write to info and debug, with info to console, LogType 3 write to info and debug, with info and debug to console.

11. matchAppend true or false. When true records are appended to the matchReport file. Usually the first cycle of accountMatch is matchAppend = false and subsequent cycles matchAppend = true.

12. MatchJoin, the schema attribute flag join is not used in accountMatch so that subsequent runs of accountMatch can use the same Schema file. The property allows multiple joins to be defined. The format is <identityMetaAttribute1>:<accountMetaAttribute1>,<identityMetaAttribute2>:<accountMetaAttribute2> etc. For example:

    ```
    GivenName:Firstname,Surname:Lastname
    ```

13. matchLogFile the name of the file that will be written with details of matches and mismatches. Matches can be checked and mismatches may indicate parameters for subsequent runs.

14. matchReport the name of the file that will be written with matches using the matchTemplate

15. matchTemplate a line containing account attributes and the identityUnique attribute. This can be used to create a batch file to update accounts after they have been matched. The template substitutes account meta attribute names with the record's actual value. It also uses one identity attribute %identityUnique% as defined in the property file. This attribute will be used in identitySync's schema as the join attribute. For example if identityUnique = employeeID, this template will update the Active Directory:

    ```
    matchTemplate = dsquery -samid %accountName% | dsmod user -empid = %identityUnique%
    ```

After the Active Directory has been updated, the Active Directory schema for identitySync will have an entry:

```
employeeID, employeeID;Join
```

To update the comment or GECOS field of a Unix account:

```
matchTemplate = usermod -c  "%firstName%
%lastName%,%location%,%telephoneNumber%,%mobile%,%IdentityUnique%" %accountName%
```

accountMatch reads CSV files and uses the CSV adapter for both the file of identities and accounts. The adapter names are:

- IdCSV for the CSV file of identities
- CSV for the CSV file of accounts

For example:

```
IdCSVresource1 = Identity1.csv
```

The other adapter properties used by accountMatch are:

1. Delim, the separator, default ",".

2. Name

3. Schema, file name of the resource's schema file.

Schema

The schema is defined in a file whose name is the adapter property "schema". Its format is fully described in the identitySync reference document:

[StagingAttributeName],[MetaverseAttributename[;AttributeModifier[;AttributeModifier][..]]]

StagingAttribute Name may be blank in cases when the Metaverse Attribute Value is evaluated, see below. A special case is the Metaverse Attribute "resource" which is copied from the Adapter Property "resource". See below.

Metaverse Attribute names may be blank, in which case the Staging Attribute name is used.

Attributes

Staging Attributes names are those found in resources, for example the column names in an Excel file or the Active Directory attribute names. Depending on the resource type, these may be case sensitive.

Metaverse attributes are created in databases of identities and resources by accountMatch. The Metaverse Attribute Names are the names that are given to the attributes in the Metaverse , these are case sensitive.

Attribute Modifiers

Modifiers mark a Metaverse Attribute for evaluation or flagged for special meaning. See identitySync Reference for further details.

Flags

Flags mark Meta attributes with special meanings:

1. displayName, marks this attribute for use as a record identifier when logging information or errors concerning the record. This modifier is mandatory.

Immediate Evaluation

Attribute modifiers are evaluated during the Transform.