How to Map OnPrem Active Directory users to existing Office365 Users

In some cases it could happen that customers create accounts on Office365 and activate only in a second phase the synchronization with Active Directory.

The problem of this approach is that in some cases the O365 users will not be associated with the corresponding AD Accounts and the user must manage the accounts twice (separate passwords, etc.)

Dirsync will perform a “Soft Match” of the user (http://support.microsoft.com/kb/2641663/en-us) but in some cases this will not work, for example if your cloud users are CRM-Only users without email Address.
In a recent project I had this issue and my approach was the following (if you have better ideas, please share – in our work we learn new things every day).

Could Users Preparation (before starting dirsync)

The could users UserPrincipalName was xxxx@tenantid.onmicrosoft.com and our goal is to have the logins in the format name.surname@customerdomain.com

The first step was renaming all the UPN’s to the new format using the Windows Azure Active Directory PowerShell

```
$cred=Get-Credential
Connect-MsolService-Credential $cred
```

list all of your users (or with the second command, only the licensed ones)

```
Get-MsolUser -all

Get-MsolUser -all | Where-Object { $_.isLicensed -eq $true }
```

Rename the users:

```
Set-MsolUserPrincipalName -UserPrincipalName olduser@tenant.onmicrosoft.com -
NewUserPrincipalName name.surname@customerdomain.com
```

I used a basic excel sheet in order to concatenate the string and repeat the command for every user

OnPremise User Preparation

In order to prepare the OnPrem users I checked the UserPrincipalName and the email address and (if needed) renamed those to the email address:

UPN = name.surname@customerdomain.com
email = name.surname@customerdomain.com

Manual match method

In order to match the user with the cloud user you have to set the Immutable ID of onPremise Active Directory user’s ObjectGUID to the immutableID value of the Office365 user.
To retrieve the ObjectGUID you can use the following command:

```
Ldifde -d "CN=xxx…,OU=xxx,DC=xxxx,DC=xx" -f c:\temp\exportuser1.txt
```

"CN=xxx…,OU=xxx,DC=xxxx,DC=xx" is the distinguished name of the user. You can use ADSIEdit or the AD Users & Computer (attribute editor) to find this value

In the Textfile exportuser1.txt look for the ObjectGUID. You will find a string like z2Xbu0xFTUapOeDqHRTN1A==

Then connect to Windows Azure Active Director and use the command

```
set-MsolUser -UserPrincipalName user1.surname1@customerdomain.com -
ImmutableId z2Xbu0xFTUapOeDqHRTN1A==
```

Retrive the ImmutableID for the users

Using ldifde to obtain the ObjectGUID, is not the best way if you have hundreds of users to manage.
To speed-up this process I prepared some basic scripts.

As first step I prepared a list of the users to be matched in a simple text file like this:

```
UserPrincpalName
user1.surname1@customerdomain.com
user2.surname2@customerdomain.com
user3.surname3@customerdomain.com
```

On my server (or wokstation) I installed the Quest Powershell Commands for ActiveDirectory (http://www.quest.com/powershell/activeroles-server.aspx) and imported all of the users in a list

```
$userlist
= @{}
$users=Import-Csv
-Path
c:\users.txt

foreach ($user in $users)

{

$Guid = [GUID](Get-QAdUser $user.UserPrincipalName).Guid

$bytearray = $guid.tobytearray()

$immutableID=""

$immutableID = [system.convert]::ToBase64String($bytearray)

write-host "utente: ",$user.UserPrincipalName," ImmutableID: " ,$immutableID

$userlist.add($user.UserPrincipalName,$immutableID)

}
```

Now in the $userList you have all of the data we need (check the contents typing $userlist in you powershell)

Set the Immutable ID for each user on Widows Azure with

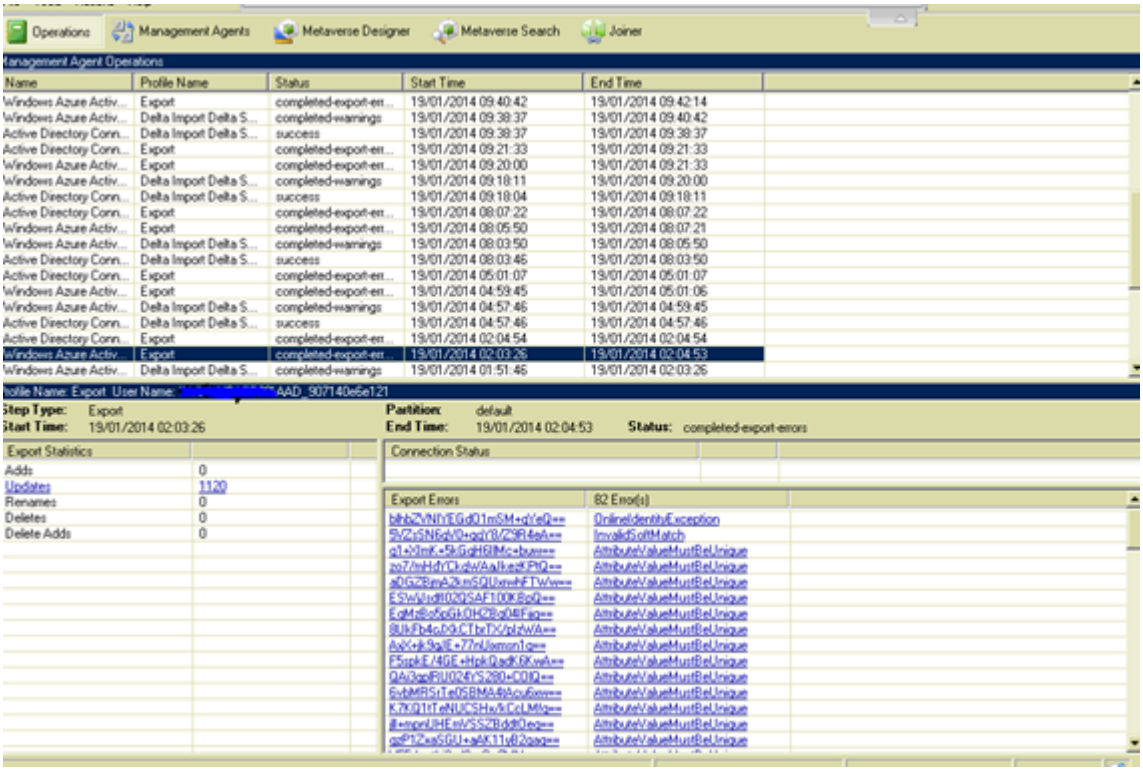
```
set-MsolUser -UserPrincipalName user1.surname1@customerdomain.com -
ImmutableId z2XbuOxFTUapOeDqHRTN1A==
```

Note: I had only 80 users to fix so I copied the ouput of the $userlist command in a textfile, added the powershell command (set-msol..) and pasted all in the powershell. Not elegant but it works.

Start the dirsync process

If you need have a little bit more details regarding the sync status, you can use the FIM shell: “C:\Program Files\Windows Azure Active Directory Sync\SYNCBUS\Synchronization Service\UIShell\miisclient.exe”

If you have, as in my case the UPN’s on Office365 and OnPrem with the same values, you will have errors like “AttributeValueMustBeUnique”.



After changing the ImmutableID as described, you should have Joins on the next Azure Import/Sync cycle:

Synchronization Statistics	
Staging	
Unchanged	520
Adds	78
Updates	40
Renames	0
Deletes	0
Inbound Synchronization	
Projections	0
Joins	78
Filtered Disconnectors	0
Disconnectors	0
Connectors with Flow Updates	78
Connectors without Flow Updates	560
Filtered Connectors	0
Deleted Connectors	0
Metaverse Object Deletes	0

Check the results on Windows Azure AD:

Before dirsync and setting the ImmutableID value:

```
PS C:\Windows\System32\WindowsPowerShell\v1.0> Get-MsolUser -UserPrincipalName
name.surname@customerdomain.com | fl

ExtensionData : System.Runtime.Serialization.ExtensionDataObject

AlternateEmailAddresses : {}

AlternateMobilePhones : {}

BlockCredential : False

Department : Operations

DisplayName : XXXXXXXXXXXXXXX

Errors :

Fax :

FirstName : XXXXXXXXXXXXXXXX

ImmutableId :

IsBlackberryUser : False

IsLicensed : True

LastDirSyncTime :

LastName : XXXXXXXXXXXX

LicenseReconciliationNeeded : False

Licenses : {tenant:CRMSTANDARD}

OverallProvisioningStatus : Success

PasswordNeverExpires : False

PreferredLanguage : en-US

ProxyAddresses : {}

SoftDeletionTimestamp :

StrongAuthenticationMethods : {}

StrongAuthenticationRequirements : {}
```

```
StrongPasswordRequired : True

Title : XXXXXXXXXXXXXXXXXXXXXXX

UsageLocation : IT

UserPrincipalName : name.surname@customerdomain.com

ValidationStatus : Healthy
```

After dirsync (immutableID setted before dirsync):

```
PS C:\Windows\System32\WindowsPowerShell\v1.0> Get-MsolUser -UserPrincipalName
name.surname@customerdomain.com | fl

ExtensionData : System.Runtime.Serialization.ExtensionDataObject

AlternateEmailAddresses : {}

AlternateMobilePhones : {}

……

FirstName : XXXXXXXXXXXXXXX

ImmutableId : ESWjJsdfIO2QSAF1OOKBpQ==

IsBlackberryUser : False

IsLicensed : True

LastDirSyncTime : 19/01/2014 11:03:07

LastName : XXXXXXXXXXX

LicenseReconciliationNeeded : False

Licenses : {tenant:CRMSTANDARD}

OverallProvisioningStatus : Success

PasswordNeverExpires : False
```