

Microsoft 365

- [Linking Single On-Prem AD Users to Azure AD via Immutable IDs](#)
- [Linking Multiple On-Prem AD Users to Azure AD via Immutable IDs](#)

Linking Single On-Prem AD Users to Azure AD via Immutable IDs

When syncing users between Active Directory and Azure AD, you sometimes run into issues where the ImmutableID doesn't match up. That's what ties an on-prem user to their Azure AD account, and if it's missing or mismatched, the accounts won't link correctly.

I put together a simple PowerShell script to help with this. It asks for a user's UPN, checks if they exist and are enabled in both AD and Azure AD, then sets the correct ImmutableID. If there's already a conflict, it will even tell you which Azure AD user currently has that ID.

Here's the full script:

```
# Requires: ActiveDirectory module and AzureAD module
# Connect to Azure AD
Connect-AzureAD

do {
    # Prompt for the user's UPN
    $UPN = Read-Host "Enter the UPN of the user to process (or type 'exit' to quit)"
    if ([string]::IsNullOrEmpty($UPN) -or $UPN -eq 'exit') { break }

    # Get the AD user by UPN
    $ADUser = Get-ADUser -Filter { UserPrincipalName -eq $UPN } -Properties ObjectGUID, Enabled

    if (-not $ADUser) {
        Write-Warning "No AD user found with UPN $UPN"
        continue
    }

    if (-not $ADUser.Enabled) {
        Write-Warning "The AD user $UPN is disabled"
        continue
    }
}
```

```

# Lookup the Azure AD user directly
$AzureUser = Get-AzureADUser -Filter "userPrincipalName eq '$UPN'"

if (-not $AzureUser) {
    Write-Warning "No matching Azure AD user found for $UPN"
    continue
}

# Convert ObjectGUID to Base64 for ImmutableID
$ImmutableID = [System.Convert]::ToBase64String($ADUser.ObjectGUID.ToByteArray())

# Try to set ImmutableID in Azure AD
try {
    Set-AzureADUser -ObjectId $AzureUser.ObjectId -ImmutableId $ImmutableID
    Write-Host "Linked AD user $UPN to Azure AD user $($AzureUser.UserPrincipalName)"
}
catch {
    Write-Warning ("Failed to set ImmutableID for {0}: {1}" -f $UPN, $_.Exception.Message)

    # Attempt to parse conflicting ObjectId from error message
    if ($_.Exception.Message -match '([0-9a-fA-F-]{36})') {
        $ConflictingId = $matches[1]
        $ConflictingUser = Get-AzureADUser -ObjectId $ConflictingId -ErrorAction
        SilentlyContinue

        if ($ConflictingUser) {
            Write-Warning "ImmutableID conflict: Already assigned to
$($ConflictingUser.UserPrincipalName) ($($ConflictingUser.DisplayName))"
        }
        else {
            Write-Warning "ImmutableID conflict with ObjectId $ConflictingId (user not
found or deleted)."

```


Linking Multiple On-Prem AD Users to Azure AD via Immutable IDs

Manually fixing ImmutableIDs one user at a time can get old fast. If you've got a whole OU of users to sync up, it's much easier to handle them in bulk.

This PowerShell script does exactly that. You pick an OU, it grabs all the enabled AD users inside, checks for matching Azure AD accounts, and then sets the correct ImmutableID. To make it quicker, it even caches all Azure AD users into memory so lookups are fast.

Here's the full script:

```
# Requires: ActiveDirectory module and AzureAD module
# Connect to Azure AD
Connect-AzureAD

# Prompt to select an OU
$OUs = Get-ADOrganizationalUnit -Filter * | Select-Object -Property Name, DistinguishedName
$OUSelection = $OUs | Out-GridView -Title "Select an OU to process" -PassThru

if (-not $OUSelection) {
    Write-Host "No OU selected, exiting."
    exit
}

# Cache all Azure AD users once into a hashtable for O(1) lookups
Write-Host "Fetching all Azure AD users..."
$AllAzureUsers = Get-AzureADUser -All $true | Select-Object ObjectId, UserPrincipalName
$AzureUserLookup = @{}
foreach ($user in $AllAzureUsers) {
    $AzureUserLookup[$user.UserPrincipalName.ToLower()] = $user
}
Write-Host "Cached $($AllAzureUsers.Count) Azure AD users."

# Get all enabled AD users in the selected OU
Write-Host "Fetching AD users from OU: $($OUSelection.Name)"
```

```

Get-ADUser -SearchBase $OUSelection.DistinguishedName -Filter * -Properties ObjectGUID,
UserPrincipalName, Enabled |
    Where-Object { $_.Enabled -eq $true } |
    ForEach-Object {
        $UPN = $_.UserPrincipalName
        Write-Host "Processing AD user: $UPN"

        # Lookup in hashtable
        $AzureUser = $AzureUserLookup[$UPN.ToLower()]

        if (-not $AzureUser) {
            Write-Warning "No matching 365 user found for $UPN"
            return
        }

        # Convert ObjectGUID to Base64 for ImmutableID
        $ImmutableID = [System.Convert]::ToBase64String($_.ObjectGUID.ToByteArray())

        # Set ImmutableID in Azure AD
        try {
            Set-AzureADUser -ObjectId $AzureUser.ObjectId -ImmutableId $ImmutableID
            Write-Host "Linked AD user $UPN to 365 user $($AzureUser.UserPrincipalName)"
        }
        catch {
            Write-Warning ("Failed to set ImmutableID for {0}: {1}" -f $UPN,
$_Exception.Message)
        }
    }

Write-Host "Done processing OU: $($OUSelection.Name)"

```