**Standardized New User Creations**

**General information:**

I’m writing this software with the intent to make my life easier. The overall idea is that you enter basic user information and the software creates the account for you. How? This software is for all intents and purposes just a front end for pushing out a series of powershell commands. It’s a little more complicated than that of course but that’s the general idea. You enter the information and I blast the account into your AD through powershell. Let’s break it down:

1. The first thing that happens when you run this software is the population of the combo boxes. We pull in your organizational unit names, distinguished names, UPNs, domain names, and all of your user accounts. There is another field for “cloud clients” as well. The UPN field is only visible to the user if a secondary UPN such as \*.com exists. If not, the widget is hidden from view. The same thing applies to the cloud clients; except it’s handled by a checkbox on the menubar.
2. Once you fill in the appropriate fields, we create the user account. As you fill in these fields, you’re literally building out a powershell command. I add some extra stuff to the mix though. The account is automatically enabled by default, the user has a proxyAddress set based on your input and the email is added to the account info tab.

To see the commands that are used just scroll till you see “Powershell command(s) used: “.

**Usage:**

**Powershell Command(s) used:**

1. **Get-ADOrganizationalUnit -Filter \* | select-object -expandproperty Name**

Note that it specifically pulls name. This is just to make it “readable”.

1. **Get-ADOrganizationalUnit -Filter \* | Select-Object -ExpandProperty distinguishedName**

Note that it specifically pulls the distinguished name. We use this later to move the user.

1. **Get-ADForest | Select-Object -ExpandProperty UPNSuffixes**

If no suffixes are returned, we don’t care.

1. **Get-ADDomain | Select-Object -expandproperty Forest**
2. **New-ADUser -Name <full name> -GivenName <given name> -Surname <surname> - AccountPassword <password> -UserPrincipalName <upn> -DisplayName <display name> -EmailAddress <email> -SamAccountName <user> -Enabled 1**

You no doubt see that there is a little more to it, but the other commands are just basic things like splits or joins to change how the name is formatted. There are still a few things that happen though.

1. **$usr = Get-ADUser -Filter {name -like $template\_account}**

1. **$groups = (Get-ADUser $usr -Properties MemberOf).MemberOf**
2. **Add-ADGroupMember -Identity (Get-ADGroup $group).name -Members $user\_name**
3. **Set-ADUser -Identity $user\_name -Add @{Proxyaddresses = $\_proxy}**
4. **Shift-OU -Username (Get-ADUser -Filter {SamAccountName -like $user\_name}) -OU\_Array (Get-ADOrganizationalUnit -Filter \* | Select-Object -expandproperty Name) -OU\_Path (Get-ADOrganizationalUnit -Filter \* | Select-Object -ExpandProperty distinguishedName)**

The Shift-OU user is a function that I built out. The powershell script is available here as well. If you want more in-depth information check that out.