

Microsoft.PowerShell_profile.ps1

```
1 # Values to use
2 $IsAdmin = (New-Object Security.Principal.WindowsPrincipal ([
Security.Principal.WindowsIdentity]::GetCurrent())).IsInRole([
Security.Principal.WindowsBuiltinRole]::Administrator);
3 $CmdPromptUser = [Security.Principal.WindowsIdentity]::GetCurrent();
4 $username = $CmdPromptUser.Name.split("\")[1]
5 $Date = Get-Date -Format "hh:mm:ss tt";
6
7 # Get current git branch
8 function Get-GitBranchName {
9 $branch = git branch 2>$null | Select-String -Pattern "^*" | ForEach-
Object { $_.ToString().Replace("* ", "").Replace("`n", "") }
10 if ($branch) {
11     "$branch"
12 }
13 }
14
15 Write-Host "Welcome to PowerShell $" ;
16
17 Clear-Host;
18 function prompt {
19 $CurFolderName = Split-Path -Path $pwd -Leaf;
20
21 # Write-Host "[$Date] " -ForegroundColor DarkGray;
22
23 if ($IsAdmin) {
24     $PermColor = "Cyan";
25     $Host.UI.RawUI.WindowTitle = "$CurFolderName | PowerShell (Admin)"
26 }
27 else {
28     $PermColor = "Yellow";
29     $Host.UI.RawUI.WindowTitle = "$CurFolderName | PowerShell"
30
31     Write-Host "[User]" -NoNewline -ForegroundColor $PermColor;
32 }
33 $branch = Get-GitBranchName;
34 if ($branch) {
35     Write-Host " ($branch)" -NoNewline -ForegroundColor DarkGray;
36 }
37
38 Write-Host " " -NoNewline;
39 Write-Host "$($CmdPromptUser.Name.split("
\")[1])" -NoNewline
40
41 Write-Host " $ " -NoNewline -ForegroundColor $PermColor
42 Write-Host "$pwd" -NoNewline
43
44 Write-Host ">" -NoNewline -ForegroundColor $PermColor;
45 return " ";
```

NewUser.ps1

```
1 $_name = Read-Host "Group Name";
2 $distName = (Get-ADDomain).DistinguishedName;
3 $domain = (Get-ADDomain).NetBIOSName;
4     function Get-OUContainer {
5     param (
6         [string[]] $OU,
7         [string[]] $CN
8     )
9
10    $CNTText = "";
11    foreach ($item in $CN) {
12        $CNTText += "CN=$item," ;
13    }
14    $OUTText = "";
15    foreach ($item in $OU) {
16        $OUTText += "OU=$item," ;
17    }
18    return "$CNTText$OUTText$distName" ;
19 }
20
21     function Create ([string] $name) {
22     $path = (Get-OUContainer);
23     if (!(Get-ADOrganizationalUnit -Filter "Name -like 'Test Users'")) {
24         New-ADOrganizationalUnit `
25             -Name "Test Users" `
26             -Path "$path" `
27             -ProtectedFromAccidentalDeletion $False
28     }
29
30     $path = (Get-OUContainer -OU "Test Users");
31
32     if ((Get-ADOrganizationalUnit -Filter "Name -like '$name'" )) {
33         Write-Warning "OU and potentially groupset already exists: $name" ;
34         Exit;
35     }
36     else {
37         New-ADOrganizationalUnit `
38             -Name "$name" `
39             -Path "$path" `
40             -ProtectedFromAccidentalDeletion $False
41     }
42     $path = (Get-OUContainer -OU "$name" , "Test Users");
43
44     # Global
45     New-ADGroup -Name "$name" `
46         -GroupCategory "Security" `
47         -GroupScope "Global" `
48         -Path "$path" ;
49
50     # Modify
51     $M = "ACL-$name-M"
52     New-ADGroup -Name "$M" `
53         -GroupCategory "Security" `
54         -GroupScope "DomainLocal" `
55         -Path "$path" ;
56
57     # Read & Execute
58     $RX = "ACL-$name-RX" ;
```

```

59 New-ADGroup -Name "$RX" `
60     -GroupCategory "Security" `
61     -GroupScope "DomainLocal" `
62     -Path "$path" ;
63
64 $user = "USR-$name" ;
65 New-ADUser -Name "$user" -Path "$path" ;
66
67 # Combine groups
68     Add-ADGroupMember -Identity "ACL-$name-M" -Members "$name"
69     Add-ADGroupMember -Identity "ACL-$name-RX" -Members "$name"
70     # Add user to group
71     Add-ADGroupMember -Identity "$name" -Members "USR-$name"
72
73     mkdir "D:\Firm\$name" ;
74
75 # Apply folder permission
76     icacls "D:\Firm\$name" /grant $domain\ACL-$name-M:(OI)`(CI`)M
77     icacls "D:\Firm\$name" /grant $domain\ACL-$name-RX:(OI)`(CI`)RX
78
79 }
80
81 Create -Name "$_name" ;
82     Write-Output "";
83     Write-Output ("User and groups stored in: " + (Get-OUContainer -OU "
$name" , "Test Users" -CN "$user" ));
84     Write-Output (" " + (Get-OUContainer -OU "
$name" , "Test Users" -CN "$RX" ));
85     Write-Output (" " + (Get-OUContainer -OU "
$name" , "Test Users" -CN "$M" ));
86     Write-Output ("User folder is stored in: " + ("D:\Firm\$name" ));

```