

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('&quot;\&quot;')[1]
5 $Date = Get-Date -Format &quot;hh:mm:ss tt&quot;;
6
7 # Get current git branch
8 function Get-GitBranchName {
9 $branch = git branch 2>$null | Select-String -Pattern &quot;^\*&quot;
| ForEach-Object { $_.ToString().Replace(&quot;* &quot;, &quot;&quot;
).Replace(&quot;\n&quot;, &quot;&quot;) }
10 if ($branch) {
11     &quot;$branch&quot;;
12 }
13}
14
15 Write-Host &quot;Welcome to PowerShell $&quot;;
16
17 Clear-Host;
18 function prompt {
19 $CurFolderName = Split-Path -Path $pwd -Leaf;
20
21 # Write-Host &quot;[$Date] &quot; -ForegroundColor DarkGray;

22 if ($IsAdmin) {
23     $PermColor = &quot;Cyan&quot;;
24     $Host.UI.RawUI.WindowTitle = &quot;$CurFolderName | PowerShell
(Admin)&quot;;
25     Write-Host &quot;[Admin]&quot; -NoNewline -ForegroundColor
$PermColor;
26 }
27 else {
28     $PermColor = &quot;Yellow&quot;;
29     $Host.UI.RawUI.WindowTitle = &quot;$CurFolderName | PowerShell&quot;;

30     Write-Host &quot;[User]&quot; -NoNewline -ForegroundColor $PermColor
;
31 }
32
33 $branch = Get-GitBranchName;
34 if ($branch) {
35     Write-Host &quot;($branch)&quot; -NoNewline -ForegroundColor DarkGray;
36 }
37
38 Write-Host &quot; &quot; -NoNewline;
39 Write-Host &quot;$($CmdPromptUser.Name.split('&quot;\&quot;')[1])&quot; -
NoNewline
40     Write-Host &quot;$ &quot; -NoNewline -ForegroundColor $PermColor

41     Write-Host &quot;$pwd&quot; -NoNewline
42
43 Write-Host &quot;>&quot; -NoNewline -ForegroundColor $PermColor;
44 return &quot; &quot;;
45}
```

NewUser.ps1

```
1$_name = Read-Host &quot;Group Name&quot;;
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    $CNTText = &quot;&quot;;
11    foreach ($item in $CN) {
12        $CNTText += &quot;CN=$item,&quot;;
13    }
14    $OUTText = &quot;&quot;;
15    foreach ($item in $OU) {
16        $OUTText += &quot;OU=$item,&quot;;
17    }
18    return &quot;$CNTText$OUTText$distName&quot;;
19}
20
21     function Create ([string] $name) {
22     $path = (Get-OUContainer);
23     if (!(Get-ADOrganizationalUnit -Filter &quot;Name -like &#x27;Test
Users&#x27;&quot;)) {
24         New-ADOrganizationalUnit `
25         -Name &quot;Test Users&quot; `
26         -Path &quot;$path&quot; `
27         -ProtectedFromAccidentalDeletion $False
28     }
29
30     $path = (Get-OUContainer -OU &quot;Test Users&quot;);
31
32     if ((Get-ADOrganizationalUnit -Filter &quot;Name -like &#x27;$name
&#x27;&quot;)) {
33         Write-Warning &quot;OU and potentially groupset already exists: $name
&quot;;
34         Exit;
35     }
36     else {
37         New-ADOrganizationalUnit `
38         -Name &quot;$name&quot; `
39         -Path &quot;$path&quot; `
40         -ProtectedFromAccidentalDeletion $False
41     }
42     $path = (Get-OUContainer -OU &quot;$name&quot;,&quot;Test Users&quot;);
43
44     # Global
45     New-ADGroup -Name &quot;$name&quot; `
46     -GroupCategory &quot;Security&quot; `
47     -GroupScope &quot;Global&quot; `
48     -Path &quot;$path&quot;;
49
50     # Modify
51     $M = &quot;ACL-$name-M&quot;;
52     New-ADGroup -Name &quot;$M&quot; `
53     -GroupCategory &quot;Security&quot; `
54     -GroupScope &quot;DomainLocal&quot; `
55     -Path &quot;$path&quot;;
```

```

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-$name
";
72
73 mkdir "D:\Firm\<u>$name</u>";
74
75 # Apply folder permission
76 icacls "D:\Firm\<u>$name</u>" /grant $domain\ACL-$name-M
: `(OI)` `(CI)` M
77 icacls "D:\Firm\<u>$name</u>" /grant $domain\ACL-$name-RX
: `(OI)` `(CI)` RX
78
79}
80
81 Create -Name "$<u>_name</u>";
82 Write-Output "";
83 Write-Output ("User and groups stored in: " + (Get-OUContainer
-OU "$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 "$M"));
86 Write-Output ("User folder is stored in: " + ("D:\Firm\
<u>$name</u>"));

```