Microsoft.PowerShell_profile.ps1

```
# Values to use
  $IsAdmin = (New-Object Security.Principal.WindowsPrincipal ([Security.Principal.WindowsIdentity]::GetCurrent())).IsInRole([Security.Principal.WindowsBuiltinRole]::Administrator);
  $CmdPromptUser = [Security.Principal.WindowsIdentity]::GetCurrent();
  $username = $CmdPromptUser.Name.split(&quot;\&quot;)[1]
  $Date = Get-Date -Format &quot;hh:mm:ss tt&quot;;

  # Get current git branch
     function Get-GitBranchName                  {
 $branch = git branch 2&gt;$null | Select-String -Pattern &quot;^\*&quot; | ForEach-Object { $_.ToString().Replace(&quot;* &quot;, &quot;&quot;).Replace(&quot;`n&quot;, &quot;&quot;) }
 if ($branch) {
   &quot;$branch&quot;
    }
}

  Write-Host &quot;Welcome to PowerShell $&quot;      ;

  Clear-Host;
     function prompt       {
 $CurFolderName = Split-Path -Path $pwd -Leaf;

 # Write-Host &quot;[$Date] &quot; -ForegroundColor DarkGray;

    if ($IsAdmin) {
   $PermColor = &quot;Cyan&quot;;
   $Host.UI.RawUI.WindowTitle = &quot;$CurFolderName | PowerShell (Admin)&quot;;
      Write-Host &quot;[Admin]&quot; -NoNewline -ForegroundColor $PermColor;
 }
 else {
   $PermColor = &quot;Yellow&quot;;
   $Host.UI.RawUI.WindowTitle = &quot;$CurFolderName | PowerShell&quot;;

      Write-Host &quot;[User]&quot; -NoNewline -ForegroundColor $PermColor ;
 }

 $branch = Get-GitBranchName;
 if ($branch) {
   Write-Host &quot; ($branch)&quot;        -NoNewline -ForegroundColor DarkGray;
 }

 Write-Host &quot; &quot; -NoNewline;
 Write-Host &quot;$($CmdPromptUser.Name.split(&quot;                  \&quot;)[1])&quot; -NoNewline
    Write-Host &quot; $ &quot;        -NoNewline -ForegroundColor $PermColor
    Write-Host &quot;$pwd&quot;       -NoNewline

 Write-Host &quot;&gt;&quot; -NoNewline -ForegroundColor $PermColor;
 return &quot; &quot;;
}
```

NewUser.ps1

```
$_name = Read-Host &quot;Group Name&quot;;
   $distName = (Get-ADDomain).DistinguishedName;
   $domain = (Get-ADDomain).NetBIOSName;
      function Get-OUContainer                {
  param (
    [string[]] $OU,
    [string[]] $CN
     )

  $CNText = &quot;&quot;;
  foreach ($item in $CN) {
    $CNText += &quot;CN=$item,&quot;       ;
  }
  $OUText = &quot;&quot;;
  foreach ($item in $OU) {
    $OUText += &quot;OU=$item,&quot;       ;
  }
  return &quot;$CNText$OUText$distName&quot;      ;
}

      function Create ([string] $name)   {
  $path = (Get-OUContainer);
  if (!(Get-ADOrganizationalUnit -Filter &quot;Name -like &#x27;Test Users&#x27;&quot;)) {
    New-ADOrganizationalUnit `
    -Name &quot;Test Users&quot; `
    -Path &quot;$path&quot;        `
    -ProtectedFromAccidentalDeletion $False
     }

  $path = (Get-OUContainer -OU &quot;Test Users&quot;);

  if ((Get-ADOrganizationalUnit -Filter &quot;Name -like &#x27;$name &#x27;&quot;            )) {
    Write-Warning &quot;OU and potentially groupset already exists: $name &quot;      ;
    Exit;
  }
  else {
    New-ADOrganizationalUnit `
      -Name &quot;$name&quot;       `
      -Path &quot;$path&quot;       `
      -ProtectedFromAccidentalDeletion $False
     }
  $path = (Get-OUContainer -OU &quot;$name&quot;      ,&quot;Test Users&quot;);

  # Global
     New-ADGroup -Name &quot;$name&quot;       `
    -GroupCategory &quot;Security&quot; `
    -GroupScope &quot;Global&quot; `
    -Path &quot;$path&quot;      ;

  # Modify
     $M = &quot;ACL-$name-M&quot;
     New-ADGroup -Name &quot;$M&quot;       `
    -GroupCategory &quot;Security&quot; `
    -GroupScope &quot;DomainLocal&quot; `
    -Path &quot;$path&quot;      ;

  # Read &amp; Execute
     $RX = &quot;ACL-$name-RX&quot;         ;
  New-ADGroup -Name &quot;$RX&quot;        `
    -GroupCategory &quot;Security&quot; `
    -GroupScope &quot;DomainLocal&quot; `
    -Path &quot;$path&quot;      ;

  $user = &quot;USR-$name&quot;      ;
  New-ADUser -Name &quot;$user&quot;       -Path &quot;$path&quot;      ;

  # Combine groups
     Add-ADGroupMember -Identity &quot;ACL-$name-M&quot;         -Members &quot;$name&quot;
     Add-ADGroupMember -Identity &quot;ACL-$name-RX&quot;          -Members &quot;$name&quot;
     # Add user to group
     Add-ADGroupMember -Identity &quot;$name&quot;       -Members &quot;USR-$name&quot;

     mkdir &quot;D:\Firm\$name&quot;      ;

  # Apply folder permission
     icacls &quot;D:\Firm\$name&quot;       /grant $domain\ACL-$name-M :`(OI`)`(CI`)M
  icacls &quot;D:\Firm\$name&quot;       /grant $domain\ACL-$name-RX :`(OI`)`(CI`)RX

}

Create -Name &quot;$_name&quot;      ;
   Write-Output &quot;&quot;;
   Write-Output (&quot;User and groups stored in: &quot; + (Get-OUContainer -OU &quot;$name&quot;      ,&quot;Test Users&quot; -CN &quot;$user&quot;));
   Write-Output (&quot;                           &quot; + (Get-OUContainer -OU &quot;$name&quot;      ,&quot;Test Users&quot; -CN &quot;$RX&quot;));
   Write-Output (&quot;                           &quot; + (Get-OUContainer -OU &quot;$name&quot;      ,&quot;Test Users&quot; -CN &quot;$M&quot;));
   Write-Output (&quot;User folder is stored in:  &quot; + (&quot;D:\Firm\$name&quot;      ));
```