How to Create New Active Directory Users with PowerShell

The easiest way to create a new user in an Active Directory domain is using the Active Directory Users and Computers MMC snap-in. However, what if you need to create multiple user accounts in bulk, or ADUC is not available for some reason? In this article, we explain several ways to create Active Directory user accounts with PowerShell using the New-ADUser cmdlet.

Create New User Accounts Using the New-Aduser Cmdlet

So what is the PowerShell cmdlet used to create user objects? It's the New-ADUser cmdlet, which is included in the Active Directory PowerShell module built into Microsoft Windows Server 2008R2/2012 and above. Therefore, the fi thing we need to do is enable the AD module:

```
Import-Module ActiveDirectory
```

Now let's take a closer look at cmdlet New-ADUser. We can get its full syntax by running the following command:

```
Get-Command New-ADUser –Syntax
```

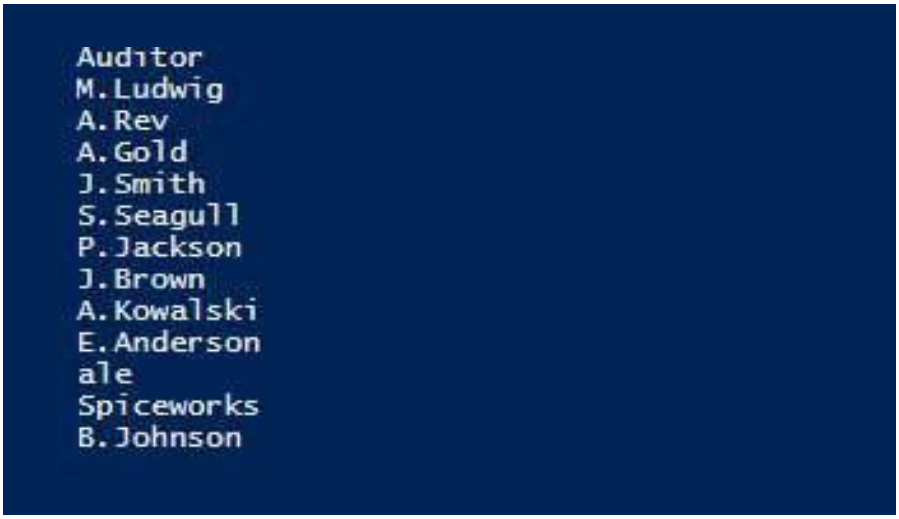
```
PS C:\Users\t.simpson> Get-Command New-ADUser -Syntax

New-ADUser [-Name] <string> [-WhatIf] [-Confirm] [-AccountExpirationDate <datetime>] [-AccountNotDelegat
ed <bool>] [-AccountPassword <securestring>] [-AllowReversiblePasswordEncryption <bool>] [-Authenticatio
nPolicy <ADAuthenticationPolicy>] [-AuthenticationPolicySilo <ADAuthenticationPolicySilo>] [-AuthType <A
DAuthType>] [-CannotChangePassword <bool>] [-Certificates <X509Certificate[]>] [-ChangePasswordAtLogon <
bool>] [-City <string>] [-Company <string>] [-CompoundIdentitySupported <bool>] [-Country <string>] [-Cr
edential <pscredential>] [-Department <string>] [-Description <string>] [-DisplayName <string>] [-Divisi
on <string>] [-EmailAddress <string>] [-EmployeeID <string>] [-EmployeeNumber <string>] [-Enabled <bool>
] [-Fax <string>] [-GivenName <string>] [-HomeDirectory <string>] [-HomeDrive <string>] [-HomePage <stri
ng>] [-HomePhone <string>] [-Initials <string>] [-Instance <ADUser>] [-KerberosEncryptionType <ADKerbero
sEncryptionType>] [-LogonWorkstations <string>] [-Manager <ADUser>] [-MobilePhone <string>] [-Office <st
ring>] [-OfficePhone <string>] [-Organization <string>] [-OtherAttributes <hashtable>] [-OtherName <stri
ng>] [-PassThru] [-PasswordNeverExpires <bool>] [-PasswordNotRequired <bool>] [-Path <string>] [-POBox <
string>] [-PostalCode <string>] [-PrincipalsAllowedToDelegateToAccount <ADPrincipal[]>] [-ProfilePath <s
tring>] [-SamAccountName <string>] [-ScriptPath <string>] [-Server <string>] [-ServicePrincipalNames <st
ring[]>] [-SmartcardLogonRequired <bool>] [-State <string>] [-StreetAddress <string>] [-Surname <string>
] [-Title <string>] [-TrustedForDelegation <bool>] [-Type <string>] [-UserPrincipalName <string>] [<Comm
onParameters>]
```

When you know the syntax, it's easy to add users to Active Directory:

```
New-ADUser B.Johnson
```

Now let’s check whether the user was added successful- ly by listing all Active Directory users using the following script:

```
Get-ADUser -Filter * -Properties samAccount- Name | select samAccountName
```



There it is, the last one in the list!

Create a New Active Directory User
Account with Password
Accounts are created with the following default properties:

- Account is created in the "Users" container.
- Account is disabled.
- Account is a member of Domain Users group.
- No password is set.
- User must reset the password at the fi logon.

Therefore, to make a new account that's actually usable, we need to enable it using the Enable-ADAccount cmdlet andgiveitapasswordusingtheSet-ADAccountPassword cmdlet.

So let's create a new account with the following attributes:

- **Name** – Jack Robinson
- **Given Name** – Jack
- **Surname** – Robinson
- **Account Name** – J.Robinson
- **User Principal Name**–J.Robinson@enterprise.com
- **Path address** – "OU=Managers,DC=enterprise,DC=com
- **Password Input**
- **Status** – Enabled

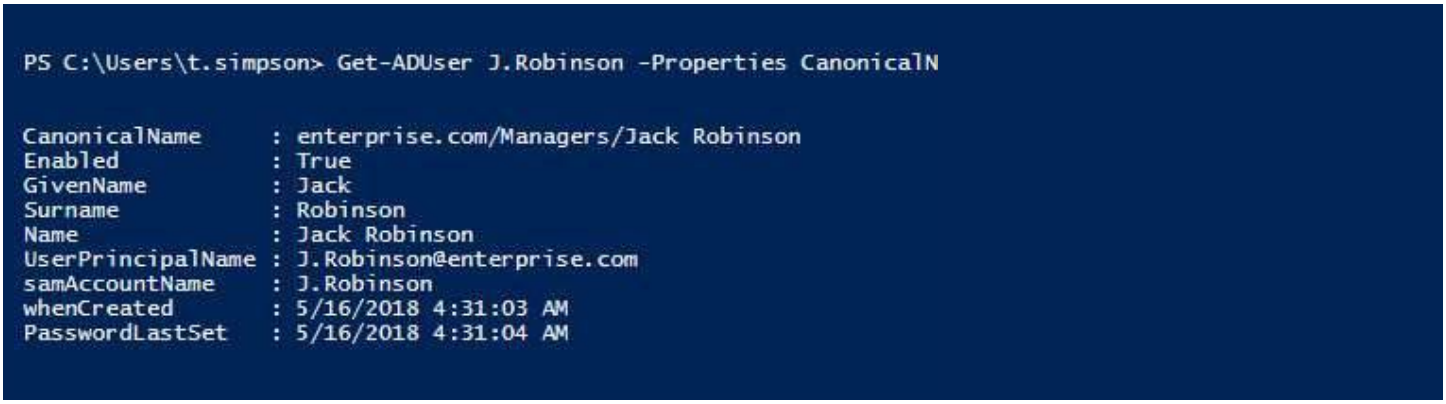
Here's the script we'll use:

```
New-ADUser -Name "Jack Robinson" -GivenName "Jack" -Surname "Robinson" -SamAccountName
"J.Robinson" -UserPrincipalName "J.Robinson@ enterprise.com" -Path "OU=Managers,DC=en-
terprise,DC=com" -AccountPassword(Read-Host -AsSecureString "Input Password") –Enabled
$true
```

The Read-Host parameter will ask you to input new pass- word. Note that the password should meet the length, complexity and history requirements of your domain se- curity policy.

Now let's take a look at the results by running the following cmdlet:

```
Get-ADUser J.Robinson -Properties Canoni- calName, Enabled, GivenName, Surname, Name,
UserPrincipalName, samAccountName, whenCre- ated, PasswordLastSet    | Select Canonical-
Name, Enabled, GivenName, Surname, Name, UserPrincipalName, samAccountName, whenCre- ated,
PasswordLastSet
```



```
PS C:\Users\t.simpson> $path="OU=IT,DC=enterprise,DC=com"
$username=Read-Host "Enter name"
$n=Read-Host "Enter Number"
$count=1..$n
foreach ($i in $count)
{ New-AdUser -Name $username$i -Path $path -Enabled $True -ChangePasswordAtLogon $true
-AccountPassword (ConvertTo-SecureString "P@ssw0rd" -AsPlainText -force) -passThru }
Enter name: ITguest
Enter Number: 5


DistinguishedName : CN=ITguest1,OU=IT,DC=enterprise,DC=com
Enabled           : True
GivenName         :
Name              : ITguest1
ObjectClass       : user
ObjectGUID        : c547a42f-f18d-448b-9a58-2c8b1239bdbd
SamAccountName    : ITguest1
SID               : S-1-5-21-611411812-3804293928-1670731417-1187
Surname           :
UserPrincipalName :

DistinguishedName : CN=ITguest2,OU=IT,DC=enterprise,DC=com
Enabled           : True
GivenName         :
Name              : ITguest2
ObjectClass       : user
ObjectGUID        : ab437e2c-c126-4514-b2ac-ed6d99bcc4d7
SamAccountName    : ITguest2
SID               : S-1-5-21-611411812-3804293928-1670731417-1188
Surname           :
```

Create AD Users in Bulk with a PowerShell SCript

Now, let's make our task a little bit harder and create ten similar Active Directory accounts in bulk, for example, for our company's IT class, and set a default password (P@ssw0rd) for each of them. To send the default password in a protected state, we must use the ConvertTo-SecureString parameter. Here's the script to use:

```
$path="OU=IT,DC=enterprise,DC=com"
$username="ITclassuser"
$count=1..10
foreach ($i in $count)
{ New-AdUser -Name $username$i -Path $path -Enabled $True -ChangePasswordAtLogon $true
-AccountPassword (ConvertTo-SecureString "P@ssw0rd" -AsPlainText -force) -passThru }
```

Now let's make our script more fl by adding the Read- Host parameter, which will ask for the name and number of users:

```
path="OU=IT,DC=enterprise,DC=com"
$username=Read-Host "Enter name"
$n=Read-Host "Enter Number"
$count=1..$n
foreach ($i in $count)
{
      New-AdUser -Name $username$i -Path $path -Enabled $True -ChangePasswordAtLogon $true
-AccountPassword (ConvertTo-SecureString "P@ ssw0rd" -AsPlainText -force) -passThru

}
```

```
PS C:\Users\t.simpson> $path="OU=IT,DC=enterprise,DC=com"
$username=Read-Host "Enter name"
$n=Read-Host "Enter Number"
$count=1..$n
foreach ($i in $count)
{ New-AdUser -Name $username$i -Path $path -Enabled $True -ChangePasswordAtLogo
-AccountPassword (ConvertTo-SecureString "P@sswOrd" -AsPlainText -force) -passT
Enter name: ITguest
Enter Number: 5

DistinguishedName : CN=ITguest1,OU=IT,DC=enterprise,DC=com
Enabled           : True
GivenName         :
Name              : ITguest1
ObjectClass       : user
ObjectGUID        : c547a42f-f18d-448b-9a58-2c8b1239bdbd
SamAccountName    : ITguest1
SID               : S-1-5-21-611411812-3804293928-1670731417-1187
Surname           :
UserPrincipalName :

DistinguishedName : CN=ITguest2,OU=IT,DC=enterprise,DC=com
Enabled           : True
GivenName         :
Name              : ITguest2
ObjectClass       : user
ObjectGUID        : ab437e2c-c126-4514-b2ac-ed6d99bcc4d7
SamAccountName    : ITguest2
SID               : S-1-5-21-611411812-3804293928-1670731417-1188
Surname           :
```

Import AD Users from a CSV File

Another option for creating users in AD is to import them from a CSV file. This option is great when you have a list of users with predefined personal details such as:

- FirstName
- LastName
- Username
- Department
- Password
- OU

The CSV file must be in UTF8 encoding and contain contact data that looks like this:

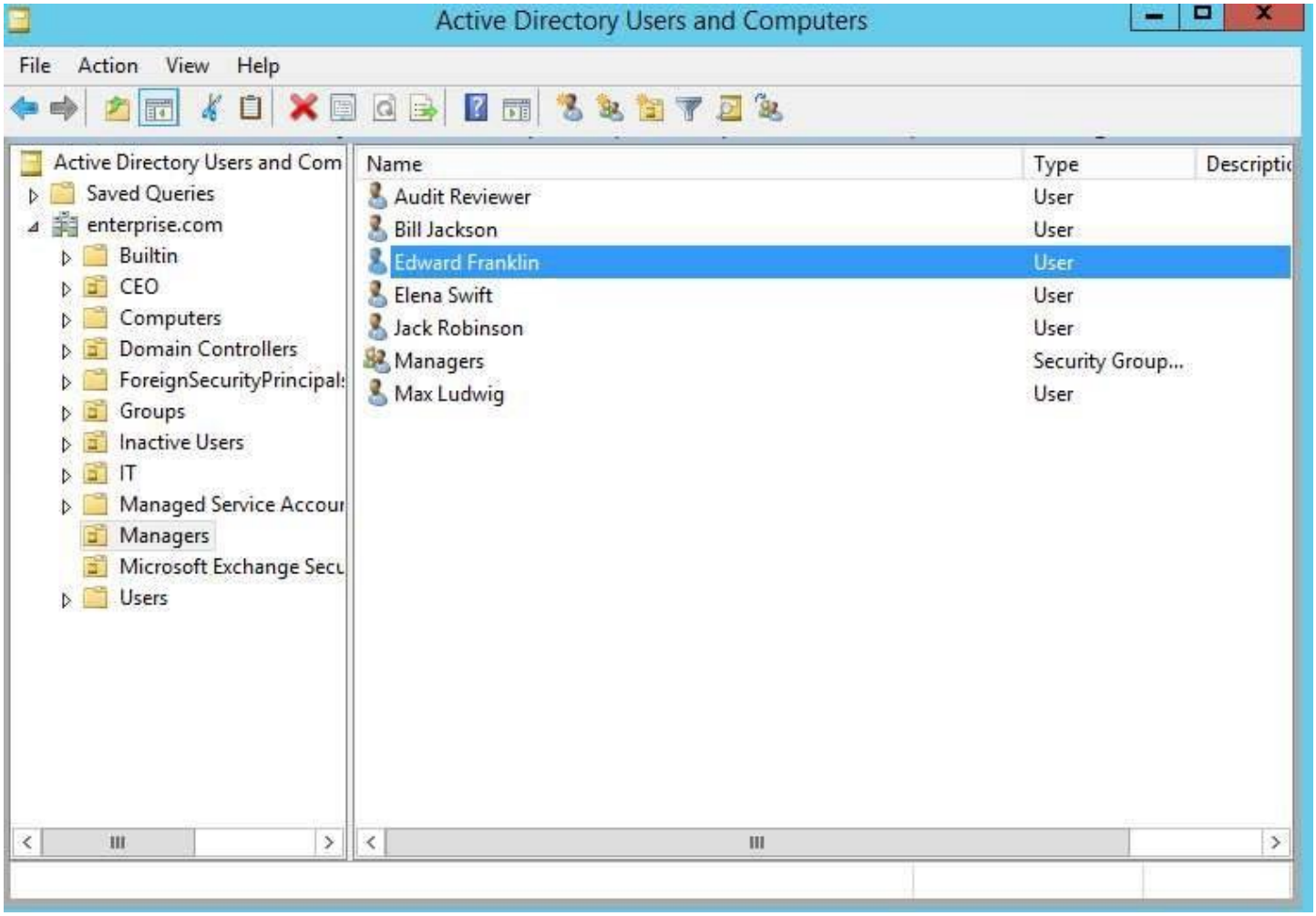
	A	B	C	D	E	F	G	H	I
1	firstname	lastname	username	department	password	ou			
2	Edward	Franklin	E.Franklin	Sales	P@s$w0rd	OU=Managers,DC=enterprise,DC=com			
3	Bill	Jackson	B.Jackson	HR	P@s$w0rd	OU=Managers,DC=enterprise,DC=com			
4									
5									
6									
7									
8									

The following script will create enabled user objects for any users in the CSV that don't already have accounts in AD. The "Reset pass- word at the next logon" option will be enabled for the new accounts, so you can use your default password:

```
#Enter a path to your import CSV fi
$ADUsers = Import-csv C:\scripts\newusers.csv
```

```
foreach ($User in $ADUsers)
{

$Username = $User.username
$Password= $User.password
$Firstname = $User.fi
$Lastname = $User.lastname
$Department = $User.department
#Check if the user account already exists in AD
      if (Get-ADUser -F {SamAccountName –eq $Username})
      {
            #If user does exist, output a  warning message
           Write-Warning "A user account $Username has already exist in Active Direc-  tory."
      }
   else
      {
            #If a user does not exist then create a new user account

            #Account will be created in the OU listed in the $OU variable in the CSV
      file; don’t forget to change the domain name in the"-UserPrincipalName" variable

New-ADUser `
    -SamAccountName $Username `
    -UserPrincipalName "$Username@ yourdomain.com" `
    -Name "$Firstname $Lastname" `
    -GivenName $Firstname `
    -Surname $Lastname `
    -Enabled $True `
    -ChangePasswordAtLogon $True `
    -DisplayName "$Lastname, $First-name" `
    -Department $Department `
    -Path $OU `
    -AccountPassword (convertto-se- curestring $Password -AsPlainText -Force)

      }
}
```

After script execution, we have two new users, Edward Franklin and Bill Jackson, in our Active Directory domain:



Let's take a look at their details by running Get-ADUser cmdlet again:

```
Get-ADUser E.Franklin -Properties Canoni- calName, Enabled, GivenName, Surname, Name,
UserPrincipalName, samAccountName, whenCreated, PasswordLastSet | Select Canonical- Name,
Enabled, GivenName, Surname, Name, UserPrincipalName, samAccountName, whenCreated,
PasswordLastSet
```