MyWebUniversity

python PowerShell FPDF Library PDF generator

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Add-Computer'

PS C:\Users\wahid> Get-Help Add-Computer

```
NAME
    Add-Computer

SYNOPSIS
    Add the local computer to a domain or workgroup.


SYNTAX
    Add-Computer [-DomainName] <System.String> [-ComputerName <System.String[]>]
    [-Credential <System.Management.Automation.PSCredential>] [-Force]
    [-LocalCredential <System.Management.Automation.PSCredential>] [-NewName
    <System.String>] [-Options {AccountCreate | Win9XUpgrade | UnsecuredJoin |
    PasswordPass | DeferSPNSet | JoinWithNewName | JoinReadOnly | InstallInvoke}]
    [-OUPath <System.String>] [-PassThru] [-Restart] [-Server <System.String>]
    [-UnjoinDomainCredential <System.Management.Automation.PSCredential>]
    [-Unsecure] [-Confirm] [-WhatIf] [<CommonParameters>]

    Add-Computer [-WorkgroupName] <System.String> [-ComputerName
    <System.String[]>] [-Credential <System.Management.Automation.PSCredential>]
    [-Force] [-LocalCredential <System.Management.Automation.PSCredential>]
    [-NewName <System.String>] [-PassThru] [-Restart] [-Confirm] [-WhatIf]
```

[<CommonParameters>]

DESCRIPTION

The `Add-Computer` cmdlet adds the local computer or remote computers to a domain or workgroup, or moves them from one domain to another. It also creates a domain account if the computer is added to the domain without an account.

You can use the parameters of this cmdlet to specify an organizational unit (OU) and domain controller or to perform an unsecure join.

To get the results of the command, use the Verbose and PassThru parameters.

PARAMETERS

-ComputerName <System.String[]>

Specifies the computers to add to a domain or workgroup. The default is the local computer.

Type the NetBIOS name, an Internet Protocol (IP) address, or a fully qualified domain name of each of the remote computers. To specify the local computer, type the computer name, a dot (`.`), or "localhost".

This parameter does not rely on Windows PowerShell remoting. You can use the ComputerName parameter of `Add-Computer` even if your computer is not configured to run remote commands.

This parameter is introduced in Windows PowerShell 3.0.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to join the computers to a new domain. The default is the current user.

Type a user name, such as "User01" or "Domain01\User01", or enter a PSCredential object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, you will be prompted for a password.

To specify a user account that has permission to remove the computer from its current domain, use the UnjoinDomainCredential parameter. To specify a user account that has permission to connect to a remote computer, use the LocalCredential parameter.

-DomainName <System.String>

Specifies the domain to which the computers are added. This parameter is required when adding the computers to a domain.

-Force <System.Management.Automation.SwitchParameter>

Suppresses the user confirmation prompt. Without this parameter, `Add-Computer` requires you to confirm the addition of each computer.

This parameter is introduced in Windows PowerShell 3.0.

-LocalCredential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to connect to the computers that are specified by the ComputerName parameter. The default is the current user.

Type a user name, such as "User01" or "Domain01\User01", or enter a PSCredential object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, you will be prompted for a password.

To specify a user account that has permission to add the computers to a new domain, use the Credential parameter. To specify a user account that has permission to remove the computers from their current domain, use the UnjoinDomainCredential parameter.

This parameter is introduced in Windows PowerShell 3.0.

-NewName <System.String>

Specifies a new name for the computer in the new domain. This parameter is valid only when one computer is being added or moved.

This parameter is introduced in Windows PowerShell 3.0.

-Options <Microsoft.PowerShell.Commands.JoinOptions>

Specifies advanced options for the `Add-Computer` join operation. Enter one or more values in a comma-separated string.

The acceptable values for this parameter are:

- AccountCreate : Creates a domain account. The `Add-Computer` cmdlet automatically creates a domain account when it adds a computer to a domain. This option is included for completeness.

- Win9XUpgrade : Indicates that the join operation is part of a Windows operating system upgrade.

- UnsecuredJoin : Performs an unsecured join. To request an unsecured join, use the Unsecure parameter or this option. If you want to pass a machine password, then you must use this option in combination with `PasswordPass` option.

- PasswordPass : Sets the machine password to the value of the Credential (DomainCredential) parameter after performing an unsecured join. This option also indicates that the value of the Credential (DomainCredential) parameter is a machine password, not a user password. This option is valid only when the `UnsecuredJoin` option is specified. When using this option, the credential provided to the `-Credential` parameter must have a null username.

- JoinWithNewName : Renames the computer name in the new domain to the name specified by the NewName parameter. When you use the NewName parameter, this option is set automatically. This option is designed to be used with the Rename-Computer cmdlet. If you use the `Rename-Computer` cmdlet to rename the computer, but do not restart the computer to make the change effective, you can use this parameter to join the computer to a domain with its new name.

- JoinReadOnly : Uses an existing machine account to join the computer to a read-only domain controller. The machine account must be added to the allowed list for password replication policy and the account password must be replicated to the read-only domain controller prior to the join operation.

- InstallInvoke : Sets the create (0x2) and delete (0x4) flags of the FJoinOptions parameter of the JoinDomainOrWorkgroup method. For more information about the JoinDomainOrWorkgroup method, see JoinDomainOrWorkgroup method of the Win32_ComputerSystem class (/windows/win32/cimwin32prov/joindomainorworkgroup-method-in-class-win32-computersystem). For more information about these options, see NetJoinDomain function (/windows/win32/api/lmjoin/nf-lmjoin-netjoindomain).

This parameter was introduced in Windows PowerShell 3.0.

-OUPath <System.String>

Specifies an organizational unit (OU) for the domain account. Enter the full distinguished name of the OU in quotation marks. The default value is the default OU for machine objects in the domain.

-PassThru <System.Management.Automation.SwitchParameter>

Returns an object representing the item with which you are working. By default, this cmdlet does not generate any output.

-Restart <System.Management.Automation.SwitchParameter>

Restarts the computers that were added to the domain or workgroup. A restart is often required to make the change effective.

This parameter is introduced in Windows PowerShell 3.0.

-Server <System.String>

Specifies the name of a domain controller that adds the computer to the domain. Enter the name in DomainName\ComputerName format. By default, no domain controller is specified.

-UnjoinDomainCredential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to remove the computers from their current domains. The default is the current user.

Type a user name, such as "User01" or "Domain01\User01", or enter a PSCredential object, such as one generated by the `Get-Credential` cmdlet. If you type a user name, you will be prompted for a password.

Use this parameter when you are moving computers to a different domain. To specify a user account that has permission to join the new domain, use the Credential parameter. To specify a user account that has permission to connect to a remote computer, use the LocalCredential parameter.

This parameter is introduced in Windows PowerShell 3.0.

-Unsecure <System.Management.Automation.SwitchParameter>

Performs an unsecure join to the specified domain.

-WorkgroupName <System.String>

Specifies the name of a workgroup to which the computers are added. The default value is "WORKGROUP".

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug, ErrorAction, ErrorVariable, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

Example 1: Add a local computer to a domain then restart the computer

```
Add-Computer -DomainName Domain01 -Restart
```

This command adds the local computer to the Domain01 domain and then restarts the computer to make the change effective.

-------- Example 2: Add a local computer to a workgroup --------

```
Add-Computer -WorkgroupName WORKGROUP-A
```

This command adds the local computer to the Workgroup-A workgroup.

--------- Example 3: Add a local computer to a domain ---------

```
Add-Computer -DomainName Domain01 -Server Domain01\DC01 -PassThru -Verbose
```

This command adds the local computer to the Domain01 domain by using the Domain01\DC01 domain controller.

The command uses the PassThru and Verbose parameters to get detailed information about the results of the command.

Example 4: Add a local computer to a domain using the OUPath parameter

```
Add-Computer -DomainName Domain02 -OUPath
"OU=testOU,DC=domain,DC=Domain,DC=com"
```

This command adds the local computer to the Domain02 domain. It uses the OUPath parameter to specify the organizational unit for the new accounts.

Example 5: Add a local computer to a domain using credentials

```
Add-Computer -ComputerName Server01 -LocalCredential Server01\Admin01
-DomainName Domain02 -Credential Domain02\Admin02 -Restart -Force
```

This command adds the Server01 computer to the Domain02 domain. It uses the LocalCredential parameter to specify a user account that has permission to connect to the Server01 computer. It uses the Credential parameter to specify a user account that has permission to join computers to the domain. It uses the Restart parameter to restart the computer after the join operation completes and the Force parameter to suppress user confirmation messages.

----- Example 6: Move a group of computers to a new domain -----

```
Add-Computer -ComputerName Server01, Server02, localhost -DomainName Domain02
-LocalCredential Domain01\User01 -UnjoinDomainCredential Domain01\Admin01
-Credential Domain02\Admin01 -Restart
```

This command moves the Server01 and Server02 computers, and the local computer, from Domain01 to Domain02.

It uses the LocalCredential parameter to specify a user account that has permission to connect to the three affected computers. It uses the UnjoinDomainCredential parameter to specify a user account that has permission to unjoin the computers from the Domain01 domain and the Credential parameter to specify a user account that has permission to join the computers to the Domain02 domain. It uses the Restart parameter to restart all three computers after the move is complete.

Example 7: Move a computer to a new domain and change the name of the computer

```
Add-Computer -ComputerName Server01 -DomainName Domain02 -NewName Server044
-Credential Domain02\Admin01 -Restart
```

This command moves the Server01 computer to the Domain02 and changes the machine name to Server044.

The command uses the credential of the current user to connect to the Server01 computer and unjoin it from its current domain. It uses the Credential parameter to specify a user account that has permission to join the computer to the Domain02 domain.

-- Example 8: Add computers listed in a file to a new domain --

```
Add-Computer -ComputerName (Get-Content Servers.txt) -DomainName Domain02
-Credential Domain02\Admin02 -Options Win9xUpgrade  -Restart
```

This command adds the computers that are listed in the `Servers.txt` file to the Domain02 domain. It uses the Options parameter to specify the Win9xUpgrade option. The Restart parameter restarts all of the newly added computers after the join operation completes.

Example 9: Add a computer to a domain using predefined computer credentials

```
New-ADComputer -Name "Server02" -AccountPassword (ConvertTo-SecureString
-String 'TempJoinPA$$' -AsPlainText -Force)

# Then this command is run from `Server02` which is not yet domain-joined:

$joinCred = New-Object pscredential -ArgumentList ([pscustomobject]@{
    UserName = $null
    Password = (ConvertTo-SecureString -String 'TempJoinPA$$' -AsPlainText
-Force)[0]
})
Add-Computer -Domain "Domain03" -Options UnsecuredJoin,PasswordPass
-Credential $joinCred
```

This combination of commands creates a new computer account with a predefined name and temporary join password in a domain using an existing domain-joined computer. Then separately, a computer with the predefined name joins the domain using only the computer name and the temporary join password. The predefined password is only used to support the join operation and is replaced as part of normal computer account procedures after the computer completes the join.

REMARKS

To see the examples, type: "get-help Add-Computer -examples".

For more information, type: "get-help Add-Computer -detailed".

For technical information, type: "get-help Add-Computer -full".

For online help, type: "get-help Add-Computer -online"