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

PowerShell Get-Help on command 'Install-Package'

PS C:\Users\wahid> Get-Help Install-Package

NAME

Install-Package

SYNOPSIS

Installs one or more software packages.

SYNTAX

Install-Package [-AcceptLicense] [-AllowClobber] [-AllowPrereleaseVersions]
[-AllVersions] [-Command <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-DscResource <System.String[]>]
[-Filter <System.String>] [-Force] [-ForceBootstrap] [-Includes {Cmdlet |
DscResource | Function | RoleCapability | Workflow}] [-InstallUpdate]
[-NoPathUpdate] [-PackageManagementProvider <System.String>] [-Proxy
<System.Uri>] [-ProxyCredential <System.Management.Automation.PSCredential>]
[-PublishLocation <System.String>] [-RoleCapability <System.String[]>] [-Scope
{CurrentUser | AllUsers}] [-ScriptPublishLocation <System.String>]
[-ScriptSourceLocation <System.String>] [-SkipPublisherCheck] [-Tag
<System.String[]>] [-Type {Module | Script | All}] [-Confirm] [-WhatIf]
[<CommonParameters>]

Install-Package [-AcceptLicense] [-AllowClobber] [-AllowPrereleaseVersions]
[-AllVersions] [-Command <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-DscResource <System.String[]>]
[-Filter <System.String>] [-Force] [-ForceBootstrap] [-Includes {Cmdlet |
DscResource | Function | RoleCapability | Workflow}] [-InstallUpdate]
[-NoPathUpdate] [-PackageManagementProvider <System.String>] [-Proxy
<System.Uri>] [-ProxyCredential <System.Management.Automation.PSCredential>]
[-PublishLocation <System.String>] [-RoleCapability <System.String[]>] [-Scope
{CurrentUser | AllUsers}] [-ScriptPublishLocation <System.String>]
[-ScriptSourceLocation <System.String>] [-SkipPublisherCheck] [-Tag
<System.String[]>] [-Type {Module | Script | All}] [-Confirm] [-WhatIf]
[<CommonParameters>]

Install-Package [-AdditionalArguments <System.String[]>] [-AllVersions]
[-Credential <System.Management.Automation.PSCredential>] [-Force]
[-ForceBootstrap] [-Proxy <System.Uri>] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-Confirm] [-WhatIf]
[<CommonParameters>]

Install-Package [-AdditionalArguments <System.String[]>] [-AllVersions]
[-Credential <System.Management.Automation.PSCredential>] [-Force]
[-ForceBootstrap] [-Proxy <System.Uri>] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-Confirm] [-WhatIf]
[<CommonParameters>]

Install-Package [-AllowPrereleaseVersions] [-AllVersions] [-ConfigFile
<System.String>] [-Contains <System.String>] [-Credential
<System.Management.Automation.PSCredential>] [-Destination <System.String>]
[-ExcludeVersion] [-FilterOnTag <System.String[]>] [-Force] [-ForceBootstrap]
[-Headers <System.String[]>] [-Proxy <System.Uri>] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-Scope {CurrentUser | AllUsers}]
[-SkipDependencies] [-SkipValidate] [-Confirm] [-WhatIf] [<CommonParameters>]

Install-Package [-AllowPrereleaseVersions] [-AllVersions] [-ConfigFile
<System.String>] [-Contains <System.String>] [-Credential
<System.Management.Automation.PSCredential>] [-Destination <System.String>]
[-ExcludeVersion] [-FilterOnTag <System.String[]>] [-Force] [-ForceBootstrap]
[-Headers <System.String[]>] [-Proxy <System.Uri>] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-Scope {CurrentUser | AllUsers}]
[-SkipDependencies] [-SkipValidate] [-Confirm] [-WhatIf] [<CommonParameters>]

Install-Package [-AllVersions] [-Credential
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]
[-IncludeSystemComponent] [-IncludeWindowsInstaller] [-Proxy <System.Uri>]
[-ProxyCredential <System.Management.Automation.PSCredential>] [-Confirm]
[-WhatIf] [<CommonParameters>]

Install-Package [-AllVersions] [-Credential
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]
[-IncludeSystemComponent] [-IncludeWindowsInstaller] [-Proxy <System.Uri>]
[-ProxyCredential <System.Management.Automation.PSCredential>] [-Confirm]
[-WhatIf] [<CommonParameters>]

Install-Package [-InputObject]
<Microsoft.PackageManagement.Packaging.SoftwareIdentity[]> [-AllVersions]
[-Credential <System.Management.Automation.PSCredential>] [-Force]
[-ForceBootstrap] [-Proxy <System.Uri>] [-ProxyCredential
<System.Management.Automation.PSCredential>] [-Confirm] [-WhatIf]
[<CommonParameters>]

Install-Package [-Name] <System.String[]> [-AllVersions] [-Credential
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]
[-MaximumVersion <System.String>] [-MinimumVersion <System.String>]
[-ProviderName {Bootstrap | NuGet | PowerShellGet}] [-Proxy <System.Uri>]
[-ProxyCredential <System.Management.Automation.PSCredential>]
[-RequiredVersion <System.String>] [-Source <System.String[]>] [-Confirm]

[-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Install-Package` cmdlet installs one or more software packages on the local computer. If you have multiple software sources, use `Get-PackageProvider` and `Get-PackageSource` to display details about your providers. !INCLUDE [nuget-module (../../includes/nuget-module.md)]

PARAMETERS

-AcceptLicense <System.Management.Automation.SwitchParameter>

AcceptLicense automatically accepts the license agreement during installation.

-AdditionalArguments <System.String[]>

Specifies one or more additional arguments for installation.

-AllowClobber <System.Management.Automation.SwitchParameter>

Overrides warning messages about conflicts with existing commands. Overwrites existing commands that have the same name as commands being installed.

-AllowPrereleaseVersions <System.Management.Automation.SwitchParameter>

Allows the installation of packages marked as prerelease.

-AllVersions <System.Management.Automation.SwitchParameter>

`Install-Package` installs all available versions of the package. By default, only the newest version is installed.

-Command <System.String[]>

Specifies one or more commands that `Install-Package` searches.

-ConfigFile <System.String>

Specifies a path that contains a configuration file.

-Contains <System.String>

`Install-Package` gets objects if the Contains parameter specifies a value that matches any of the object's property values.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to access the computer and run commands. Type a user name, such as User01 , Domain01\User01 , or enter a PSCredential object, generated by the `Get-Credential` cmdlet. If you type a user name, you're prompted for a password.

When the Credential parameter isn't specified, `Install-Package` uses the current user.

-Destination <System.String>

Specifies a path to an input object.

-DscResource <System.String[]>

Specifies one or more Desired State Configuration (DSC) resources that are searched by `Install-Package`. Use the `Find-DscResource` cmdlet to find DSC resources.

-ExcludeVersion <System.Management.Automation.SwitchParameter>

Switch to exclude the version number in the folder path.

-Filter <System.String>

Specifies terms to search for within the Name and Description properties.

-FilterOnTag <System.String[]>

Specifies a tag that filters results and excludes results that don't contain the specified tag.

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation. Overrides restrictions that prevent `Install-Package` from succeeding, with the exception of security.

-ForceBootstrap <System.Management.Automation.SwitchParameter>

Forces PackageManagement to automatically install the package provider for the specified package.

-Headers <System.String[]>

Specifies the package headers.

-Includes <System.String[]>

Specifies whether `Install-Package` should find all package types. The acceptable values for this parameter are as follows:

- Cmdlet

- DscResource

- Function

- RoleCapability

- Workflow

-IncludeSystemComponent <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet includes system components in the results.

-IncludeWindowsInstaller <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet includes the Windows installer in the results.

-InputObject <Microsoft.PackageManagement.Packaging.SoftwareIdentity[]>

Accepts pipeline input. Specifies a package by using the package's SoftwareIdentity type. `Find-Package` outputs a SoftwareIdentity object.

-InstallUpdate <System.Management.Automation.SwitchParameter>

Indicates that `Install-Package` installs updates.

-MaximumVersion <System.String>

Specifies the maximum allowed package version that you want to install. If you don't specify this parameter, `Install-Package` installs the package's newest version.

-MinimumVersion <System.String>

Specifies the minimum allowed package version that you want to install. If you don't add this parameter, `Install-Package` installs the package's newest version that satisfies any version specified by the MaximumVersion parameter.

-Name <System.String[]>

Specifies one or more package names. Multiple package names must be separated by commas.

-NoPathUpdate <System.Management.Automation.SwitchParameter>

NoPathUpdate only applies to the `Install-Script` cmdlet. NoPathUpdate is a dynamic parameter added by the provider and isn't supported by `Install-Package`.

-PackageManagementProvider <System.String>

Specifies the name of the PackageManagement provider.

-ProviderName <System.String[]>

Specifies one or more package provider names to which to scope your package search. You can get package provider names by running the

`Get-PackageProvider` cmdlet.

-Proxy <System.Uri>

Specifies a proxy server for the request, rather than connecting directly to an internet resource.

-ProxyCredential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to use the proxy server specified by the Proxy parameter.

-PublishLocation <System.String>

Specifies the path to a package's published location.

-RequiredVersion <System.String>

Specifies the exact allowed version of the package that you want to install. If you don't add this parameter, `Install-Package` installs the package's newest version that satisfies any version specified by the MaximumVersion parameter.

-RoleCapability <System.String[]>

Specifies an array of role capabilities.

-Scope <System.String>

Specifies the scope for which to install the package. The acceptable values for this parameter are as follows:

- CurrentUser

- AllUsers

-ScriptPublishLocation <System.String>

Specifies the path to a script's published location.

-ScriptSourceLocation <System.String>

Specifies the script source location.

-SkipDependencies <System.Management.Automation.SwitchParameter>

Skips the installation of software dependencies.

-SkipPublisherCheck <System.Management.Automation.SwitchParameter>

Allows you to get a package version that is newer than your installed version. For example, an installed package that is digitally signed by a trusted publisher but a new version isn't digitally signed.

-SkipValidate <System.Management.Automation.SwitchParameter>

Switch that skips validating the credentials of a package.

-Source <System.String[]>

Specifies one or more package sources. Multiple package source names must be separated by commas. You can get package source names by running the `Get-PackageSource` cmdlet.

-Tag <System.String[]>

Specifies one or more strings to search for in the package metadata.

-Type <System.String>

Specifies whether to search for packages with a module, a script, or both. The acceptable values for this parameter are as follows:

- Module

- Script

- All

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>
Shows what would happen if `Install-Package` cmdlet is run. 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: Install a package by package name ---------

PS> Install-Package -Name NuGet.Core -Source MyNuGet -Credential Contoso\TestUser

`Install-Package` uses parameters to specify the packages Name and Source . The Credential parameter uses a domain user account with permissions to install packages. The command prompts you for the user account password.
------- Example 2: Use Find-Package to install a package -------

PS> Find-Package -Name NuGet.Core -Source MyNuGet | Install-Package

`Find-Package` uses the Name and Source parameters to locate a package. The object is sent down the pipeline and `Install-Package` installs the package on the local computer.
Example 3: Install packages by specifying a range of versions

PS> Install-Package -Name NuGet.Core -Source MyNuGet -MinimumVersion 2.8.0 -MaximumVersion 2.9.0

`Install-Package` uses the Name and Source parameters to find a package. The

MinimumVersion and MaximumVersion parameters specify a range of software versions. The highest version in the range is installed.

REMARKS

To see the examples, type: "get-help Install-Package -examples".

For more information, type: "get-help Install-Package -detailed".

For technical information, type: "get-help Install-Package -full".

For online help, type: "get-help Install-Package -online"