



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 'Save-Package'

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

NAME

Save-Package

SYNOPSIS

Saves packages to the local computer without installing them.

SYNTAX

```
Save-Package [-AcceptLicense] [-AllowPrereleaseVersions] [-AllVersions]
[-Command <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-DscResource <System.String[]>]
[-Filter <System.String>] [-Force] [-ForceBootstrap] [-Includes {DscResource |
Cmdlet | Function | Workflow | RoleCapability}] [-LiteralPath <System.String>]
[-PackageManagementProvider <System.String>] [-Path <System.String>] [-Proxy
<System.Uri>] [-ProxyCredential <System.Management.Automation.PSCredential>]
[-PublishLocation <System.String>] [-RoleCapability <System.String[]>]
[-ScriptPublishLocation <System.String>] [-ScriptSourceLocation
<System.String>] [-Tag <System.String[]>] [-Type {Module | Script | All}]
[-Confirm] [-WhatIf] [<CommonParameters>]
```

Save-Package [-AcceptLicense] [-AllowPrereleaseVersions] [-AllVersions]

[-Command <System.String[]>] [-Credential
<System.Management.Automation.PSCredential>] [-DscResource <System.String[]>]
[-Filter <System.String>] [-Force] [-ForceBootstrap] [-Includes {DscResource |
Cmdlet | Function | Workflow | RoleCapability}] [-LiteralPath <System.String>]
[-PackageManagementProvider <System.String>] [-Path <System.String>] [-Proxy
<System.Uri>] [-ProxyCredential <System.Management.Automation.PSCredential>]
[-PublishLocation <System.String>] [-RoleCapability <System.String[]>]
[-ScriptPublishLocation <System.String>] [-ScriptSourceLocation
<System.String>] [-Tag <System.String[]>] [-Type {Module | Script | All}]
[-Confirm] [-WhatIf] [<CommonParameters>]

Save-Package [-AllowPrereleaseVersions] [-AllVersions] [-ConfigFile
<System.String>] [-Contains <System.String>] [-Credential
<System.Management.Automation.PSCredential>] [-FilterOnTag <System.String[]>]
[-Force] [-ForceBootstrap] [-Headers <System.String[]>] [-LiteralPath
<System.String>] [-Path <System.String>] [-Proxy <System.Uri>]
[-ProxyCredential <System.Management.Automation.PSCredential>] [-SkipValidate]
[-Confirm] [-WhatIf] [<CommonParameters>]

Save-Package [-AllowPrereleaseVersions] [-AllVersions] [-ConfigFile
<System.String>] [-Contains <System.String>] [-Credential
<System.Management.Automation.PSCredential>] [-FilterOnTag <System.String[]>]
[-Force] [-ForceBootstrap] [-Headers <System.String[]>] [-LiteralPath
<System.String>] [-Path <System.String>] [-Proxy <System.Uri>]
[-ProxyCredential <System.Management.Automation.PSCredential>] [-SkipValidate]
[-Confirm] [-WhatIf] [<CommonParameters>]

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

```
Save-Package [-Name] <System.String[]> [-AllVersions] [-Credential
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]
[-LiteralPath <System.String>] [-MaximumVersion <System.String>]
[-MinimumVersion <System.String>] [-Path <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 `Save-Package` cmdlet saves packages to the local computer but doesn't install the packages. This cmdlet saves the newest version of a package unless you specify a RequiredVersion. The Path and LiteralPath parameters are mutually exclusive, and cannot be added to the same command.

PARAMETERS

- AcceptLicense <System.Management.Automation.SwitchParameter>
Automatically accept the license agreement during installation if the package requires it.
- AllowPrereleaseVersions <System.Management.Automation.SwitchParameter>
Allows packages marked as Prerelease to be saved.
- AllVersions <System.Management.Automation.SwitchParameter>
Indicates that this cmdlet saves all available versions of the package.
- Command <System.String[]>
Specifies one or more commands included in the package.
- ConfigFile <System.String>

Specifies a configuration File.

-Contains <System.String>

`Save-Package` gets objects if any item in the object's property values are an exact match for the specified value.

-Credential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to save a package from a specified package provider or source.

-DscResource <System.String[]>

Specifies one or more Desired State Configuration (DSC) resources for the package.

-Filter <System.String>

Specifies a filter for the package.

-FilterOnTag <System.String[]>

Specifies the tag that filters the results. Results that don't contain the specified tag are excluded.

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation.

-ForceBootstrap <System.Management.Automation.SwitchParameter>

Indicates that `Save-Package` forces PackageManagement to automatically install the package provider for the specified package.

-Headers <System.String[]>

Specifies the headers for the package.

-Includes <System.String[]>

Indicates the resources that the package includes.

-InputObject <Microsoft.PackageManagement.Packaging.SoftwareIdentity>

A software ID object that represents the package that you want to save.

Software IDs are part of the results of the `Find-Package` cmdlet.

-LiteralPath <System.String>

Specifies the literal path to which you want to save the package. You

cannot add both this parameter and the Path parameter to the same command.

-MaximumVersion <System.String>

Specifies the maximum version of the package that you want to save.

-MinimumVersion <System.String>

Specifies the minimum version of the package that you want to find.

-Name <System.String[]>

Specifies one or more package names.

-PackageManagementProvider <System.String>

Specifies a package management provider.

-Path <System.String>

Specifies the location on the local computer to store the package.

-ProviderName <System.String[]>

Specifies one or more provider names.

-Proxy <System.Uri>

Specifies a proxy server for the request, rather than a direct connection to the internet resource.

-ProxyCredential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to use the proxy server that

is specified by the Proxy parameter.

-PublishLocation <System.String>

Specifies the publish location.

-RequiredVersion <System.String>

Specifies the exact version of the package to save.

-RoleCapability <System.String[]>

Specifies an array of role capabilities.

-ScriptPublishLocation <System.String>

Specifies the script publish location.

-ScriptSourceLocation <System.String>

Specifies the script source location.

-SkipValidate <System.Management.Automation.SwitchParameter>

Switch that skips validating the credentials of a package.

-Source <System.String[]>

Specifies one or more package sources.

-Tag <System.String[]>

Specifies a tag to search for within the package metadata.

-Type <System.String>

Specifies whether to search for packages with a module, a script, or either.

-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: Save a package to the local computer -----

```
PS> Save-Package -Name NuGet.Core -ProviderName NuGet -Path C:\LocalPkg
```

Name	Version	Source	Summary
-----	-----	-----	-----
Microsoft.Web.Xdt	3.0.0	Nuget	Microsoft Xml Document Transformation (XDT) enables...
NuGet.Core	2.14.0	Nuget	NuGet.Core is the core framework assembly for NuGet...

`Save-Package` uses the Name parameter to specify the package. The package is downloaded from the repository specified by the ProviderName parameter. The Path parameter determines where the package is saved.

----- Example 2: Save a specific package version -----

```
PS> Save-Package -Name NuGet.Core -RequiredVersion 2.9.0 -ProviderName NuGet -Path C:\LocalPkg
```

Name	Version	Source	Summary
-----	-----	-----	-----
Microsoft.Web.Xdt	3.0.0	Nuget	Microsoft Xml Document Transformation (XDT) enables...
NuGet.Core	2.9.0	Nuget	NuGet.Core is the core framework

assembly for NuGet...

`Save-Package` uses the Name parameter to specify the package. RequiredVersion indicates a specific package version. The package is downloaded from the repository specified by the ProviderName parameter. The Path parameter determines where the package is saved.

----- Example 3: Use Find-Package to save a package -----

```
PS> Find-Package -Name NuGet.Core -ProviderName NuGet | Save-Package -Path  
C:\LocalPkg
```

`Find-Package` uses the Name parameter to specify the package. The package is downloaded from the repository specified by the ProviderName parameter. The object is sent down the pipeline to `Save-Package`. The Path parameter determines where the package is saved.

----- Example 4: Save and install the package -----

```
PS> Save-Package -Name NuGet.Core -ProviderName NuGet -Path C:\LocalPkg  
PS> Install-Package C:\LocalPkg\NuGet.Core.2.14.0.nupkg
```

`Save-Package` downloads the package file and its dependencies to the local computer. `Install-Package` installs the package and dependencies from the specified directory.

REMARKS

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

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

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

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