



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-PSSnapin'

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

NAME

Add-PSSnapin

SYNOPSIS

Adds one or more Windows PowerShell snap-ins to the current session.

SYNTAX

Add-PSSnapin [-Name] <System.String[]> [-PassThru] [<CommonParameters>]

DESCRIPTION

The `Add-PSSnapin` cmdlet adds registered Windows PowerShell snap-ins to the current session. After the snap-ins are added, you can use the cmdlets and providers that the snap-ins support in the current session.

To add the snap-in to all future Windows PowerShell sessions, add an `Add-PSSnapin` command to your Windows PowerShell profile. For more information, see `about_Profiles` (`about/about_Profiles.md`).

Windows PowerShell are packaged in modules. The exception is Microsoft.PowerShell.Core , which is a snap-in (PSSnapin). By default, only the Microsoft.PowerShell.Core snap-in is added to the session. Modules are imported automatically on first use and you can use the Import-Module cmdlet to import them.

PARAMETERS

`-Name <System.String[]>`

Specifies the name of the snap-in. This is the Name, not the AssemblyName or ModuleName. Wildcards are permitted.

To find the names of the registered snap-ins on your system, type

``Get-PSSnapin -Registered``.

`-PassThru <System.Management.Automation.SwitchParameter>`

Indicates that this cmdlet returns an object that represents each added snap-in. By default, this cmdlet does not generate any output.

`<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 snap-ins -----

```
PS C:\> Add-PSSnapIn -Name Microsoft.Exchange, Microsoft.Windows.AD
```

This command adds the Microsoft Exchange and Active Directory snap-ins to the current session.

----- Example 2: Add all the registered snap-ins -----

```
PS C:\> Get-PSSnapin -Registered | Add-PSSnapin -Passthru
```

This command adds all of the registered Windows PowerShell snap-ins to the session. It uses the `Get-PSSnapin` cmdlet with the `Registered` parameter to get objects representing each of the registered snap-ins. The pipeline operator (`|`) passes the result to `Add-PSSnapin``, which adds them to the session. The `PassThru` parameter returns objects that represent each of the added snap-ins.

----- Example 3: Register a snap-in and add it -----

```
PS C:\> Get-PSSnapin
PS C:\> Get-PSSnapin -Registered
PS C:\> Set-Alias installutil
$env:windir\Microsoft.NET\Framework\v2.0.50727\installutil.exe
PS C:\> installutil C:\Dev\Management\ManagementCmdlets.dll
PS C:\> Get-PSSnapin -Registered
PS C:\> add-pssnapin ManagementFeatures
PS C:\> Get-Command -Module ManagementFeatures
PS C:\> (Get-Command Set-Alias).pssnapin
```

This example demonstrates the process of registering a snap-in on your system and then adding it to your session. It uses `ManagementFeatures`, a fictitious snap-in implemented in a file that is named `ManagementCmdlets.dll`.

REMARKS

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

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

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

For online help, type: `"get-help Add-PSSnapin -online"`