



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 'Unregister-PackageSource'

PS C:\Users\wahid> Get-Help Unregister-PackageSource

NAME

Unregister-PackageSource

SYNOPSIS

Removes a registered package source.

SYNTAX

```
Unregister-PackageSource [-ConfigFile <System.String>] [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
[-SkipValidate] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
Unregister-PackageSource [-ConfigFile <System.String>] [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
[-SkipValidate] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
Unregister-PackageSource [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
-InputObject <Microsoft.PackageManagement.Packaging.PackageSource[]>  
[-Confirm] [-WhatIf] [<CommonParameters>]
```

```
Unregister-PackageSource [-Source] <System.String> [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
[-Location <System.String>] [-ProviderName {Bootstrap | NuGet |  
PowerShellGet}] [-Confirm] [-WhatIf] [<CommonParameters>]
```

```
Unregister-PackageSource [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
[-PackageManagementProvider <System.String>] [-PublishLocation  
<System.String>] [-ScriptPublishLocation <System.String>]  
[-ScriptSourceLocation <System.String>] [-Confirm] [-WhatIf]  
[<CommonParameters>]
```

```
Unregister-PackageSource [-Credential  
<System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]  
[-PackageManagementProvider <System.String>] [-PublishLocation  
<System.String>] [-ScriptPublishLocation <System.String>]  
[-ScriptSourceLocation <System.String>] [-Confirm] [-WhatIf]  
[<CommonParameters>]
```

DESCRIPTION

The `Unregister-PackageSource` cmdlet removes a registered package source. Package sources are always managed by a package provider. To find package sources, use the `Get-PackageSource` cmdlet.

PARAMETERS

`-ConfigFile <System.String>`

Specifies a configuration file.

`-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, the current user account is used.

-Force <System.Management.Automation.SwitchParameter>

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

-ForceBootstrap <System.Management.Automation.SwitchParameter>

Indicates that `Unregister-PackageSource` forces PackageManagement to automatically uninstall the package provider for the specified package source.

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

Accepts pipeline input that specifies the PackageSource object from the `Get-PackageSource` cmdlet. InputObject accepts the PackageSource object as a `Get-PackageSource` value or a variable that contains the object.

-Location <System.String>

Specifies the location to which a package source points. The value of this parameter can be a URI, a file path, or any other destination format that is supported by the package provider.

-PackageManagementProvider <System.String>

Specifies the PackageManagement provider.

-ProviderName <System.String>

Specifies the provider name.

-PublishLocation <System.String>

Specifies the publish location.

`-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.

`-Source <System.String>`

Specifies the friendly name of the package source.

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

Prompts you for confirmation before ``Unregister-PackageSource`` is run.

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

Shows what would happen if ``Unregister-PackageSource`` cmdlet is run. The cmdlet isn't 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: Unregister a package source for the NuGet provider

```
PS> Unregister-PackageSource -Source MyNuGet
```

The ``Unregister-PackageSource`` cmdlet uses the Source parameter to specify which source to remove.

Example 2: Use a PackageSource object to unregister a package

```
PS> $pkgsource = Get-PackageSource -Name MyNuGet
```

```
PS> Unregister-PackageSource -InputObject $pkgsource
```

The `$pkgsource` variable stores the PackageSource created by the `Get-PackageSource` cmdlet. `Unregister-PackageSource` uses the `$pkgsource` as input to the `InputObject` parameter.

As an alternative, the `Unregister-PackageSource` cmdlet can specify a value for the `InputObject` parameter:

```
Unregister-PackageSource -InputObject ( Get-PackageSource -Name MyNuGet )
```

REMARKS

To see the examples, type: "get-help Unregister-PackageSource -examples".

For more information, type: "get-help Unregister-PackageSource -detailed".

For technical information, type: "get-help Unregister-PackageSource -full".

For online help, type: "get-help Unregister-PackageSource -online"