



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

Windows PowerShell Get-Help on Cmdlet 'Get-AppxPackage'

PS:\>Get-HELP Get-AppxPackage -Full

NAME

Get-AppxPackage

SYNOPSIS

Gets a list of the app packages that are installed in a user profile.

SYNTAX

```
Get-AppxPackage [[-Name] <System.String>] [[-Publisher] <System.String>] [-AllUsers] [-PackageTypeFilter {None | Main  
| Framework | Resource | Bundle | Xap |  
Optional}] [-User <System.String>] [-Volume <AppxVolume>] [<CommonParameters>]
```

DESCRIPTION

The `Get-AppxPackage` cmdlet gets a list of the app packages that are installed in a user profile. An app package has an `.msix` or `.appx` file extension. To get the

list of packages for a user profile other than the profile for the current user, you must run this command with administrator permissions.

PARAMETERS

-AllUsers <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet lists app packages for all user accounts on the computer. To use this parameter, you must run the command with administrator permissions.

Required? false

Position? named

Default value False

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Name <System.String>

Specifies the name of a particular package. If you specify this parameter, the cmdlet returns results for this package only. Wildcards are permitted.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-PackageTypeFilter <PackageTypes>

Specifies one or more comma-separated types of packages that the cmdlet gets from the package repository.

By default, this cmdlet returns only packages of types `Main` and `Framework`.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-Publisher <System.String>

Specifies the publisher of a particular package. If you specify this parameter, the cmdlet returns results only for this publisher. Wildcards are permitted.

Required? false

Position? 2

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-User <System.String>

Specifies a user. If you specify this parameter, the cmdlet returns a list of app packages that are installed for only the user that this cmdlet specifies. To get

the list of packages for a user profile other than the profile for the current user, you must run this command with administrator permissions. The user name can

be in one of these formats:

- `domain\user_name`

- `user_name@fqn.domain.tld`

- `user_name`

- `SID-string`

Required? false

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Volume <AppxVolume>

Specifies an AppxVolume object. If you specify this parameter, this cmdlet returns only packages that are ~~relative~~ to

volume that this parameter specifies.

Required?	false
Position?	named
Default value	None
Accept pipeline input?	True (ByPropertyName, ByValue)
Accept wildcard characters?	false

<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>).

INPUTS

System.String[]

OUTPUTS

Microsoft.Windows.Appx.PackageManager.Commands.AppxPackage

This cmdlet returns an AppxPackage object that contains information, including the full name of the app package.

NOTES

---- Example 1: Get all app packages for every user account ----

Get-AppxPackage -AllUsers

This command lists the app packages that are installed for every user account on the computer.

----- Example 2: Get an app package for a specific a user -----

```
Get-AppxPackage -Name "Package17" -User "Contoso\EvanNarvaez"
```

This command displays information about `Package17` if it's installed in the specified user profile.

----- Example 3: Get a particular app package information -----

```
Get-AppxPackage -Name Microsoft.ScreenSketch
```

This command displays information about the ScreenSketch app.

-- Example 4: Get all app packages for a particular Publisher --

```
Get-AppxPackage -Publisher "CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington, C=US"
```

This command lists all the Microsoft app packages that are installed on the computer.

----- Example 5: Get all app packages with PackageTypeFilter -----

```
Get-AppxPackage -PackageTypeFilter Bundle,Framework,Main,Resource
```

This command lists all the app packages with PackageTypeFilter installed on the computer.

RELATED LINKS

	Online	Version:
https://learn.microsoft.com/powershell/module/appx/get-appxpackage?view=windowsserver2022-ps&wt.mc_id=ps-gethelp		
Package Manager API https://go.microsoft.com/fwlink/?LinkId=245447		
How to Add and Remove Apps https://go.microsoft.com/fwlink/?LinkID=231020		
Add-AppxPackage		
Get-AppxPackageManifest		
Move-AppxPackage		
Remove-AppxPackage		

