

# Winget: The best way to keep Windows apps updated

By Ed Tittel, Computerworld | JAN 19, 2023

The winget command, which taps into the Windows Package Manager service, makes it a breeze to keep applications up to date in Windows 10 or 11. Here's how to use it.

For a little more than a year, I've been working with *winget* daily to monitor and maintain my Windows 10 and 11 PCs. For those not already in the know, winget is the built-in, PowerShell-based interface to Microsoft's Windows Package Manager service.

Winget is designed to enable "users to discover, install, upgrade, remove and configure applications on Windows 10 and 11 computers," [according to Microsoft Learn](#). In my experience, winget is helpful for checking and updating most applications that run on Windows.

Please note: winget is included with Windows 10 version 1709 and later and all versions of Windows 11 as the App Installer. If you're running an earlier version of Windows 10, visit the [winget home page](#) at GitHub. There, click the *Latest* link under "Releases" at right, and download an item named "Microsoft.DesktopAppInstaller...msixbundle" (the missing characters identify Microsoft Store apps). Double-click on this item to install it. (Don't worry: if you do this on a newer Windows version it will inform you, "The App Installer is already installed.")

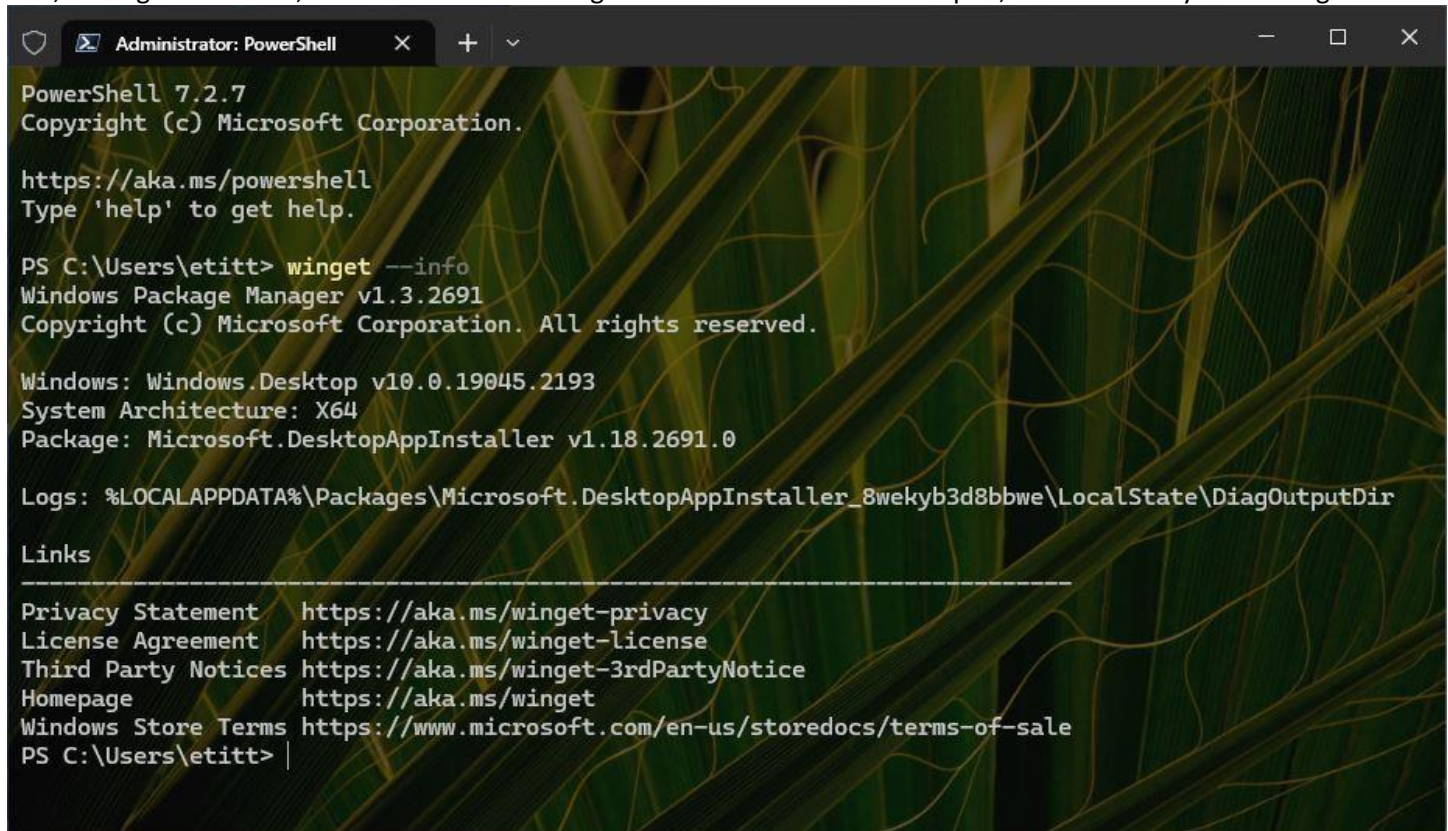
## Exploring a PC with winget

Using winget starts with opening a PowerShell window with administrator permissions: press Windows key + X and then pick *PowerShell (Admin)* or *Terminal (Admin)* from the pop-up menu. Given that winget is a PowerShell command, it uses straightforward PowerShell syntax to provide information or perform actions.

Winget tells you about itself if you enter the command:

```
winget --info
```

As shown in Figure 1, this displays the version of Windows Package Manager that's running, along with system and OS info, the log file location, and links to various winget information resources. Helpful, but not terribly interesting.



```
Administrator: PowerShell
PowerShell 7.2.7
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Users\etitt> winget --info
Windows Package Manager v1.3.2691
Copyright (c) Microsoft Corporation. All rights reserved.

Windows: Windows.Desktop v10.0.19045.2193
System Architecture: X64
Package: Microsoft.DesktopAppInstaller v1.18.2691.0

Logs: %LOCALAPPDATA%\Packages\Microsoft.DesktopAppInstaller_8wekyb3d8bbwe\LocalState\DiagOutputDir

Links
-----
Privacy Statement  https://aka.ms/winget-privacy
License Agreement  https://aka.ms/winget-license
Third Party Notices https://aka.ms/winget-3rdPartyNotice
Homepage           https://aka.ms/winget
Windows Store Terms https://www.microsoft.com/en-us/storedocs/terms-of-sale
PS C:\Users\etitt> |
```

Figure 1: The `winget --info` command shows Windows Package Manager and OS version, its supporting package ID and version, log file location, and more. (Click image to enlarge it.)

Winget also supports two information display subcommands. The `list` subcommand shows what's currently installed on the target PC. The `show` subcommand searches the online database of available package manifests to show you what information is known about resulting search hits.

With no qualifiers or queries, `winget list` shows a list of every item installed on your PC (225 items on my production PC; 218 items on one of my Windows 11 Dev Channel test PCs). It includes all standard executables and Microsoft Store apps in this count.

`Winget show` doesn't work unless you provide it with a search string of some kind. It's normally used to search for specific packages, or to see if they exist. Try it out with search strings like "windows," "power," "powershell," and so forth. You'll quickly see that it's a much more focused tool. I use it myself mostly when winget tells me a package needs an upgrade, so I can check for version numbers, publisher, and whether or not it comes from the Microsoft Store.

The `winget search` command is actually a little more helpful than `show` when looking for specific items. It will list all items that include the search string. Thus, if you use the same search strings recommended in the preceding paragraph, you'll get many more — and more interesting, usually — results back. Figure 2 shows the output from the `winget search PowerShell` command. (It includes items with PowerShell in their names, IDs, and tags, so it's much more inclusive.)

```
PS C:\Users\etitt> winget search PowerShell
Name                               Id                               Version    Match    Source
-----
PowerShell                         9MZ1SNWT0N5D                    Unknown             msstore
PowerShell Preview                 9P95ZZKTNRN4                    Unknown             msstore
PowerShell Conference Asia 2015    9WZDNCRD37D8                    Unknown             msstore
PowerShell                         Microsoft.PowerShell             7.2.7.0             winget
Windows Terminal Preview          Microsoft.WindowsTerminal.Prev... 1.16.2641.0 Tag: PowerShell winget
Windows Terminal                  Microsoft.WindowsTerminal        1.15.2874.0 Tag: powershell winget
PowerShell Preview                Microsoft.PowerShell.Preview     7.3.8.0    Tag: powershell winget
ConEmu                             Maximus5.ConEmu                  11.220.8070 Tag: powershell winget
EasyConnect                       lstratman.easyconnect            3.1.0.105  Tag: powershell winget
Oh My Posh                         JanDeDobbeleer.OhMyPosh         12.10.0    Tag: powershell winget
TfsCmdlets                         Igoravl.TfsCmdlets               2.6.0.2898 Tag: powershell winget
electerm                           electerm.electerm                1.23.28    Tag: powershell winget
wol                                 DarkfullDante.wol                1.0.2     Tag: powershell winget
AutomatedLab                       AutomatedLab.AutomatedLab        5.43.0    Tag: powershell winget
PowerSession                       Watfaq.PowerSession              0.1.7     Tag: powershell winget
PowerShell Universal Desktop       IronmanSoftware.PowerShellUniv... 2.10.2             winget
PowerShell Universal              IronmanSoftware.PowerShellUniv... 2.10.2             winget
PS C:\Users\etitt> |
```

Ed Tittel/IDG

Figure 2: Winget search PowerShell results include PowerShell items, Windows Terminal items, and related apps and applications. (Click image to enlarge it.)

## Winget's star subcommand: upgrade

My fave among the winget subcommands is the `upgrade` item. It offers insight into available updates and various ways to perform them on a Windows PC. There are three variants of `winget upgrade` most likely to be informative and useful:

```
winget upgrade
```

```
winget upgrade --all
```

```
winget upgrade --all --include-unknown
```

By itself, the `winget upgrade` command (no additional arguments or modifiers) simply shows if any newer versions of installed packages are available. Figure 3 shows an example of this command from one of my Dev Channel test PCs, with some items in need of updates. Notice that the Version column identifies the version currently installed, while the Available column identifies the corresponding update version one could apply.



Notice that you will see installer windows and the like open and close as winget runs through the various activities involved in performing those updates. Notice further that when updating web browsers — for example, Chrome — if that browser is open when you run winget, you must relaunch it manually for that update to complete. (If it's closed, it will complete on its own.)

Running `winget upgrade` again after performing those updates shows nothing left to do. That's what the cryptic message "No installed package found matching input criteria" at the bottom of Figure 4 means.

## When the upgrade command fails

Sometimes, winget updates don't clear the items that appear when you enter the `winget upgrade` command by itself. That means something remains on your PC that winget couldn't handle. By experience, I've observed the following possibilities, each of which has its own potential solution:

**Multiple copies of the same app or application are resident.** If you have multiple installations of the same program, only one is likely to be current and up to date. Unless you require older versions, the simplest fix is to uninstall them so only the current, updated version remains present.

I've seen this happen with PowerShell, for example, where some of my PCs retained version 7.2.5 even when 7.2.6 or 7.2.7 (the current version as I write this) were also present. Using Programs and Features (or some equivalent third-party tool like [Revo Uninstaller Free](#)), you can find and uninstall out-of-date versions.

**Strange programs show up you've never seen before and don't need.** Case in point: occasionally an item named "Teams machine-wide installer" shows up on my PCs. It's something Microsoft uses that apparently gets left behind from time to time. Uninstalling this item causes no noticeable issues with Teams, and it removes the item from further upgrade consideration.

**Current winget packages aren't available for some apps.** One of winget's limitations is that it can only work with items registered in its package database. You may need to visit the app publisher's website to find current updates that aren't registered with winget.

In the past, I've covered using third-party automated tools such as KC Softwares' SUMo and Patch My PC to keep apps updated in Windows 10 and 11. These and other update scanners may find items in need of updating on your PC that winget cannot handle. On my PCs, that includes applications such as Nitro Pro (a PDF reader/editor), Amazon Kindle (for which only an outdated package is available via winget), CrystalDiskMark and CrystalDiskInfo, FileZilla, various Intel tools (e.g., Intel Driver & Support Assistant), and more.

If you're willing to research your applications and their sources for updates, you can almost always find a way to get them updated. That said, winget cannot handle any and all apps on its own. Many or most of them, yes; all of them, no.

## The remove/replace maneuver

If winget can't upgrade an application, it may be able to help you anyway, with something I call a remove/replace maneuver. It means uninstalling the old version (that's the remove part) and then installing the current, up-to-date one in its place (that's the replace part).

I've run into this phenomenon with the free version of Zoom recently. The publisher removed the internal update capability inside the app. But you can use `winget uninstall Zoom.Zoom` (be careful to match its ID field exactly, which you can find easily using `winget search zoom`) to remove the current installed version. Then you can use `winget install Zoom.Zoom` to install the current, up-to-date version of the app. (Zoom maintains current winget package definitions, unlike Amazon with Kindle.)

Problem solved!

## A winget for all seasons

As you become familiar with winget, you'll find it to be a terrific tool for helping keep Windows systems (and reference or canonical Windows images for automated deployments) up to date. It's become my tool of choice for keeping apps updated because it's fast and easy to use. Although I still use SUMo to scan my systems to tell me what needs updates and Patch My PC to handle a handful of things that winget can't, winget remains my go-to tool to keep systems current. If you try it for yourself, I believe you're likely to continue using it for the very same reasons. See [Microsoft's winget documentation](#) for a complete list of commands and options.