



Netz-Weise
Lernen von den Besten.



ScriptRunner[®]
Your #1 for PowerShell Automation



au2mator

```
# > $eventData =  
@{  
    'date' = '2023-09-16 10:00'  
    'location' = 'Hannover'  
}  
# > New-PSSat @eventData
```

PowerShell Saturday

Hannover 2023

<https://pssat.de>



Eine Paketverwaltung für Windows (Eine Geschichte voller Missverständnisse)

Andreas Nick & Thorsten Butz

about_Session

Mit der AppInstaller CLI (aka WinGet) hat Microsoft die Tür zu einer Paketverwaltung für Windows aufgestoßen. Schon der eigentümliche Name scheint ein Signal zu sein, wie schwer sich Microsoft tut, seine Plattform für eine zeitgerechte Form der Softwareverwaltung neu aufzustellen. Und über der Frage des Paketmanagements schwebt die noch grundsätzlichere Frage des Paketformats: MSIX und artverwandte "modern apps" tun sich schwer, die angestaubten Klassiker wie MSI vom Thron zu stoßen.

Wir lernen mit Scott Hanselman, Keivan Beigi und Demetrius Nelon die Protagonisten eines langjährigen Feldversuchs kennen, fremde Welten zu entdecken, neue Formate und neue Plattformen zu etablieren und Wege zu beschreiten, die nie ein Admin mitgegangen ist.

Andreas Nick und Thorsten Butz werfen einen launigen Insider-Blick auf Microsofts Ringen um eine Paketverwaltung und deren langen Weg vom Nano-Server (!) über Argument Completer hin zu nativen PowerShell-Cmdlets.

<https://pssat.de/session/andreasnickthorstenbutz/>

Starring



Von links nach rechts:

Keivan Beigi

Demetrius Nelon

Scott Hanselman

Was bisher geschah ..

PackageManager (Beispiel)

- DPKG initial release: January 1994

<https://en.wikipedia.org/wiki/Dpkg>

- APT initial release: March 1998

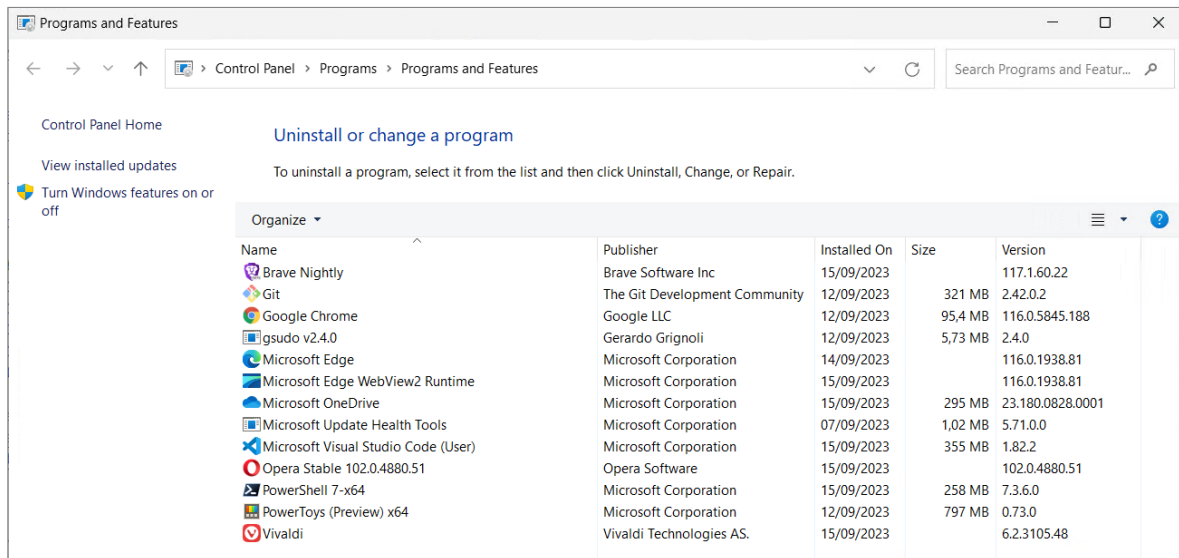
[https://en.wikipedia.org/wiki/APT_\(software\)](https://en.wikipedia.org/wiki/APT_(software))



Windows PackageManagement?



HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall*



The screenshot shows the 'Programs and Features' window in Windows. It displays a list of installed programs with columns for Name, Publisher, Installed On, Size, and Version. The list includes programs like Brave Nightly, Git, Google Chrome, gsudo v2.4.0, Microsoft Edge, Microsoft Edge WebView2 Runtime, Microsoft OneDrive, Microsoft Update Health Tools, Microsoft Visual Studio Code (User), Opera Stable 102.0.4880.51, PowerShell 7-x64, PowerToys (Preview) x64, and Vivaldi.

Name	Publisher	Installed On	Size	Version
Brave Nightly	Brave Software Inc	15/09/2023		117.1.60.22
Git	The Git Development Community	12/09/2023	321 MB	2.42.0.2
Google Chrome	Google LLC	12/09/2023	95.4 MB	116.0.5845.188
gsudo v2.4.0	Gerardo Grignoli	12/09/2023	5.73 MB	2.4.0
Microsoft Edge	Microsoft Corporation	14/09/2023		116.0.1938.81
Microsoft Edge WebView2 Runtime	Microsoft Corporation	15/09/2023		116.0.1938.81
Microsoft OneDrive	Microsoft Corporation	15/09/2023	295 MB	23.180.0828.0001
Microsoft Update Health Tools	Microsoft Corporation	07/09/2023	1.02 MB	5.71.0.0
Microsoft Visual Studio Code (User)	Microsoft Corporation	15/09/2023	355 MB	1.82.2
Opera Stable 102.0.4880.51	Opera Software	15/09/2023		102.0.4880.51
PowerShell 7-x64	Microsoft Corporation	15/09/2023	258 MB	7.3.6.0
PowerToys (Preview) x64	Microsoft Corporation	12/09/2023	797 MB	0.73.0
Vivaldi	Vivaldi Technologies AS.	15/09/2023		6.2.3105.48

```
function Get-SoftwareInventory {
    [CmdletBinding()]
    [Alias('gswi')]
    Param (
        [Parameter(Mandatory, ValueFromPipeline, ValueFromPipelineByPropertyName)]
        [string[]] $Computersname
    )
    Begin {
        $MyInvocation.Line | Write-Verbose
        $stopwatch = [System.Diagnostics.Stopwatch]::StartNew()
    }
    Process {
        ## GetSoftware, simplified
        $code = {
            $path = 'HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\*',
                'HKLM:\Software\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\*'
            Get-ItemProperty -Path $path | Where-Object -Property DisplayName
        }

        Invoke-Command -ComputerName $Computersname -ScriptBlock $code |
            Select-Object -Property DisplayVersion, DisplayName, PSComputersname #, UninstallStrin
    }
    End {
        $stopwatch.Stop()
        'Runtime(ms): ' + $stopwatch.ElapsedMilliseconds | Write-Verbose
    }
}
```

PackageManagement (aka OneGet)

What is PackageManagement (OneGet)?

OneGet is a Windows package manager, renamed as PackageManagement. It is a unified interface to package management systems and aims to make Software Discovery, Installation, and Inventory (SDII) work via a common set of cmdlets (and eventually a set of APIs).

Regardless of the installation technology underneath, users can use these common cmdlets to install/uninstall packages, add/remove/query package repositories, and query a system for the software installed.

<https://github.com/OneGet/oneget#what-is-packagemanagement-oneget>

<https://github.com/OneGet/oneget>

Announcement


This module is currently **not** in development. We are no longer accepting any pull requests to this repository. OneGet is in a stable state and is expected to receive only high-priority bug fixes from Microsoft in the future.





Alternatively, we would suggest looking at the community AnyPackage project as an actively developed replacement for OneGet.








<https://github.com/OneGet/oneget#announcement>





<https://github.com/OneGet/oneget>




OneGet




 **AndreasNick / SimplePackageProvider** Public

 Notifications  Fork 0  Star 0 


 **Code**  Issues  Pull requests  Actions  Projects  Security  Insights


 master  Go to file  Code  About


 **AndreasNick** Version for PoSh Saturday  on Oct 4, 2019  2


 SimplePackageProvi...	Version for PoSh Saturday	4 years ago
 Testpackages	Version for PoSh Saturday	4 years ago
 Test-PackageProvide...	Version for PoSh Saturday	4 years ago

No description, website, or topics provided.

 Activity

 0 stars

 2 watching

 0 forks

[Report repository](#)

<https://github.com/AndreasNick/SimplePackageProvider>

Demo A

Scoop demo

about_ModernApps

Application Virtualization



https://en.wikipedia.org/wiki/Microsoft_App-V

Windows 8

2012

Surface RT/
Surface Pro

2013

Surface 2 (RT)/
Surface Pro 2



Demo B

Windows 10 MSIX packaging environment

about_WinGet

Keivan Beigi

APPGET

Why Chocolatey is Broken Beyond Any Hope



KEIVAN BEIGI

7 OCT 2014 • 4 MIN READ

<https://keivan.io/why-chocolatey-is-broken>



APPGET

The Day AppGet Died.

The story of how Microsoft embraced and then killed AppGet.



KEIVAN BEIGI

24 MAY 2020 • 6 MIN READ

<https://keivan.io/the-day-appget-died>

<http://slidingwindows.de/slwl8>

Scott Hanselmann

Rename-Item

```
-Path AppInstallerCLI.exe  
-NewName WinGet.exe
```



Episode 20

Winget with **Demitrius Nelon**

Recorded in Praha (Prague) on June 22, 2023



vimeo

Sliding Windows, Episode 20

Demitrius about the WinGet.Client module (in June 2023)

<https://slidingwindows.de/sl20>

Begin

<https://vimeo.com/853953408#t=1h4m4s>

<https://vimeo.com/853953408/aa4f0fd6e6#t=1h4m4s>

End

1h5m47

<https://semver.org/spec/v2.0.0.html>

Semantic Versioning 2.0.0

Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes
2. MINOR version when you add functionality in a backward compatible manner
3. PATCH version when you make backward compatible bug fixes

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Semantic Versioning: Introduction

In the world of software management there exists a dreaded place called “dependency hell.” The bigger your system grows and the more packages you integrate into your software, the more likely you are to find yourself, one day, in this pit of despair.

[..]

As a solution to this problem, we propose a simple set of rules and requirements that dictate how version numbers are assigned and incremented. These rules are based on but not necessarily limited to pre-existing widespread common practices in use in both closed and open-source software. For this system to work, you first need to declare a public API. This may consist of documentation or be enforced by the code itself. Regardless, it is important that this API be clear and precise. Once you identify your public API, you communicate changes to it with specific increments to your version number. Consider a version format of X.Y.Z (Major.Minor.Patch). Bug fixes not affecting the API increment the patch version, backward compatible API additions/changes increment the minor version, and backward incompatible API changes increment the major version.

We call this system “Semantic Versioning.” Under this scheme, version numbers and the way they change convey meaning about the underlying code and what has been modified from one version to the next.

[..]

Demo C



WinGet (aka AppInstallerCLI)

about_Future

Readme

- Andreas@YT:
<https://www.youtube.com/@MADDay-pz8sb/videos>
- <https://www.nick-it.de/blog/msix/msix-und-appx-pakete-mit-powershell-von-der-kommandozeile-starten>
- <https://www.nick-it.de/blog/msix/msix-packaging-tool>
- <https://www.nick-it.de/blog/app-v/msix-tutorial-ein-umfassender-leitfaden-mit-24-kapiteln>
- <https://github.com/AndreasNick/SimplePackageProvider>
- <https://msixhero.net/>
- WinGet – behind the scenes (English video)
<https://www.thorsten-butz.de/winget-dupsug/>
- Hinter den Kulissen von WinGet (PSUG Hannover)
<https://www.thorsten-butz.de/winget/>
- Interview Demitrius Nelon (Sliding Windows 20)
<https://www.slidingwindows.de/slsw20>
- Interview Keivan Beigi (Sliding Windows 18)
<https://www.slidingwindows.de/slsw18>
- Codebeispiele, Folien
<https://github.com/thorstenbutz/conferences>
<https://github.com/AndreasNick>

<https://www.nick-it.de>

<https://www.thorsten-butz.de>

<https://www.psugh.de>

<https://pssat.de>

Versioning is not a problem *

Windows Package Manager Version numbers (2023-09-06)

winget.exe --info	v1.6.2291-preview
Get-AppxPackage	1.21.2291.0
Get-AppxProvisionedPackage	2023.817.2326.0
File version (AppInstallerCli.exe)	1.21.2308.17001
Latest stable release	1.5.2201

"Versioning is not a problem. Problems can be solved." (Jeffrey Snover)

```
winget.exe -version (short info)
```

```
winget.exe --info (detailed info)
```

```
Get-AppxPackage -Name Microsoft.DesktopAppInstaller
```

```
Get-AppxProvisionedPackage -Online |
```

```
Where-Object -Property DisplayName -eq
```

```
-Value Microsoft.DesktopAppInstaller
```

```
Get-Item "$home\AppData\Local\Microsoft\WindowsApps\*"
```