

Windows PacMan - nova igračka za sistemce



Nagradno pitanje za geekove: S koliko komandnolinjskih upravitelja instalacijskim paketima (package managers) će raspolagati Desetka od 2. kolovoza ove godine? Nećete pogriješiti odgovorite li „s previše njih, uvezvi u obzir dugogodišnju praksu“.

Instalacijom značajke *Linux subsystem for Windows*, koju nam donosi samo-što-nije *Anniversary Update*, susrest ćemo stare znance: apt, apt-* , aptitude i dpkg. Što se tiče *Powershell Package Managementa*, u osnovi agregatora nekolicine upravitelja paketima za Windows platformu, on je već na Desetki, no spomenuti *Anniversary Update* donosi unaprijedenu inačicu. Što znači da ćemo ubrzo na Desetki imati priliku iz komandne linije istovremeno (de)instaliravati aplikacije za Linux i Windowse, baš kako smo započeli na sljedećoj slici.

```
root@WIN10MSDN: ~# apt
apt 1.0.1ubuntu2 for amd64 compiled on May 18 2016 09:49:01
Usage: apt [options] command
      CLI for apt.
      Basic commands:
      list - list packages
      search - search in package descriptions
      show - show package details
      update - update lists
      install - install packages
      remove - remove packages
      upgrade - upgrade the system by installing upgrading packages
      full-upgrade - upgrading packages
      edit-sources - edit the source information file
root@WIN10MSDN: ~# PS C:\> get-command -module PackageManagement
CommandType      Name
Cmdlet           Find-Package
Cmdlet           Find-PackageProvider
Cmdlet           Get-Package
Cmdlet           Get-PackageProvider
Cmdlet           Import-PackageProvider
Cmdlet           Install-Package
Cmdlet           Install-PackageProvider
Cmdlet           Register-PackageSource
Cmdlet           Save-Package
Cmdlet           Set-PackageSource
Cmdlet           Uninstall-Package
Cmdlet           Unregister-PackageSource
```

Dodatkom Linux subsystem for Windows malo smo se zabavljali prije nešto sitno dana, vidi <https://sysportal.carnet.hr/node/1641> [1]. Junak ovog članka jer **Powershell Package Management**, za prijatelje **PacMan** ;-). Riječ je o softverskom rješenju koje objedinjuje razne postojeće upravitelje paketima i repozitorije aplikacija za Windowse. *Windows PacMan* ima svoju povijest, no nećemo se njome baviti, samo ćemo spomenuti presudan utjecaj filozofije otvorenog koda i konkretnih projekata poput **OneGet** i **Chocolatey**, dakle, ako mu je gospodin Microsoft tata, gospođa *Open-Source Community PacManu* je mama. U usporedbi sa svojom linuxoidnom sabraćom

PacMan je mali i nerazvijen, ali očekujemo njegovo naglo bildanje, barem tako nastupaju tatini trubaduri.

PacMan je prisutan u PowerShellu verzije 5, znači, sastavni je dio Desetke i Windows Servera 2016. No, spomenutu verziju PowerShella možemo dobiti i na niže verzije windoza ako na njih instaliramo **Windows Management Framework 5.0**. Znači, ne moramo imati Desetku ili WinSvr 2016 da bismo se družili s PacManom.

Na prvoj slici možemo vidjeti naredbe kojima PacMan raspolaže. Opis njihove namjene i brojnih opcija lako ćemo naći na Webu, ali treba znati da je lokalni help sustav izdašan, samo povremeno zadajte **update-help** kako biste skinuli ažurirane informacije. Kako rekosmo, rješenje se ubrzano razvija.

Da bismo si osigurali maksimalnu slobodu igranja s PacManom:

- otvarati PowerShell konzolu sa Run as Administrator
- postaviti PowerShell Execution Policy na Unrestricted

Prvo što treba napraviti je instalacija **NuGet klijenta**, jer bez njega PacMan ne može ništa konkretno odraditi. Dovoljna je naredba **find-package**, ili **find-packageprovider** pa da PacMan sam ponudi instalaciju. Svakako prihvatići.

```
Windows PowerShell
PS C:\> Find-PackageProvider
Name          Version      Source
----          -----      -----
nuget         2.8.5.205   https://oneget.org/nuget-2...
chocolatey    2.8.5.130   https://oneget.org/Chocolatey...
The provider 'nuget v2.8.5.205' is not installed.
nuget may be manually downloaded from
https://oneget.org/Microsoft.PackageManagement.NuGetProvider-2.8.5.205.dll and
installed.
Would you like PackageManagement to automatically download and install 'nuget' now?
[Y] Yes  [N] No  [S] Suspend  [?] Help (default is "Y"):
```

Od ovog trenutka možemo pretraživati defaultni repozitorij **PSGallery**. No, kad već imamo podršku za pakete **NuGet** servisa, dodat ćemo i jednu od njegovih galerija.

```
register-packageprovider -name "nuget" -location "https://www.nuget.org/api/v2"
-providername "NuGet" -trusted
```

Sada, kako vidimo s naredne slike, raspolažemo s dva repozitorija Windows aplikacija. Sudeći prema informaciji prisutnoj u koloni IsTrusted, za PacMana je repozitorij PSGallery nepouzdan, ne vjeruje mu, pa će tijekom instalacije bilo kakve aplikacije iz ovog repozitorija tražiti odobrenje.

Windows PowerShell

PS C:\> get-packagesource

Name	ProviderName	IsTrusted	IsRegistered	IsValidated
nuget	NuGet	True	True	True
PSGallery	PowerShellGet	False	True	False

PS C:\> .

To nas može ugnjaviti ako intenzivno rabimo taj repozitorij (psss, u naredbi si možemo pomoći opcijom -force), ali stvarno veliki problem postaje kad moramo instalacije odradivati programatski. Zato ćemo PacManu poručiti da je i taj repozitorij pouzdan:

set-packagesource -name "psgallery" -trusted

Idemo se na brzaka učlaniti i u javni ogrank popularnog Chocolatey servisa za distribuciju Windows aplikacija.

install-packageprovider -name "chocolatey" -force
set-packagesource -name "chocolatey" -trusted

Kako vidimo na narednoj slici, naš PacMan je sada opremljen s programskom logikom (providerima) koja mu omoguće rad s paketima raznih formata, od dobro nam poznatih exe/msi do „novog lika u kvartu“ - nupkg.

Administrator: Windows PowerShell

PS C:\> Get-PackageProvider

Name	Version	DynamicOptions
Chocolatey	2.8.5.130	SkipDependencies, ContinueOnFailure, AdditionalArguments
msi	3.0.0.0	
msu	3.0.0.0	
NuGet	2.8.5.205	Destination, ExcludeVersion, Scope, PackageManagementProvider, Type, Scope
PowerShellGet	1.0.0.1	IncludewindowsInstaller, IncludewindowsFeature
Programs	3.0.0.0	

PS C:\> .

Od sada nadalje možemo pretraživati repozitorije Chocolatey, NuGet i PSGallery naredbom find-package. Pri tome važi pravilo: ako u naredbi ne navedemo repozitorij, PacMan će tragati za aplikacijom u svima. Tako će nam naredba

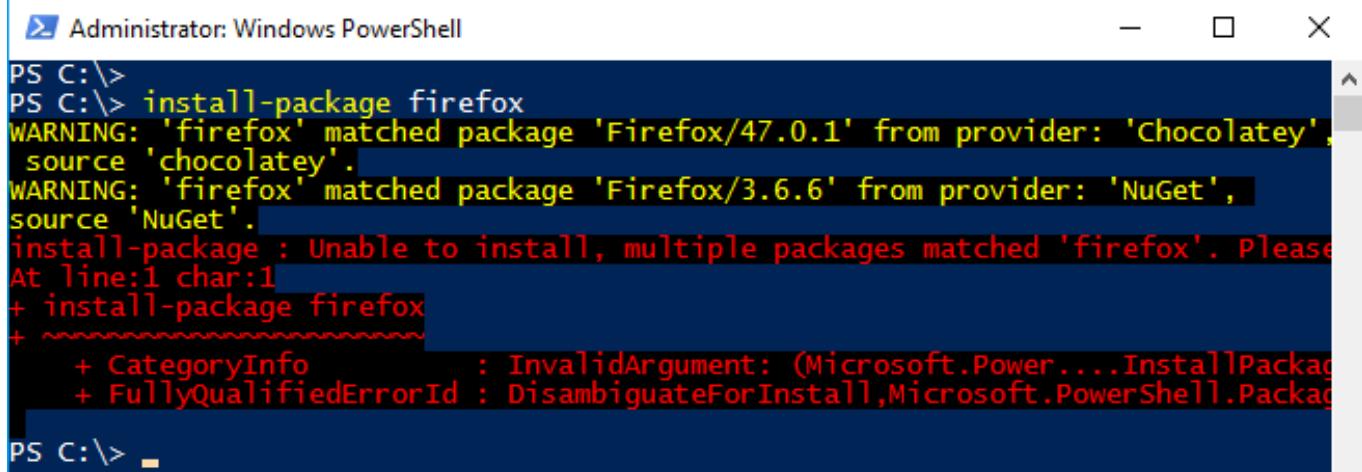
find-package -name "sysinternals"

pokazati da se takav paket nalazi u Chocolatey repozitoriju. Nažalost, do daljnega se ne možemo osloniti na uporabu jedinstvenih izraza za pretraživanje jer search mašine raznih servisa imaju vlastitu logiku filtriranja podataka. Probajte ovako i usporedite rezultate:

find-package -name "sys"
find-package -name "sys*"

U kontekstu razlika, gornji primjer je samo onaj poslovični „vrh sante leda“. Recimo, PSGallery će spremati skinute pakete u %ProgramFiles%\WindowsPowerShell, a Chocolatey u %SystemDrive%\Chocolatey, svaki ima svoj format paketa i pravila... itd. Bit će bolje, idemo mi dalje.

Radimo li instalacijsku skriptu, nije na odmet parametrom -providername navesti servis od kojega želimo preuzeti aplikaciju, kako nam se ne bi dešavale situacije poput ove na sljedećoj slici, kad su različiti paketi istog imena prisutni u više repozitorija.



```
Administrator: Windows PowerShell
PS C:\> install-package firefox
WARNING: 'firefox' matched package 'Firefox/47.0.1' from provider: 'Chocolatey', source 'chocolatey'.
WARNING: 'firefox' matched package 'Firefox/3.6.6' from provider: 'NuGet', source 'NuGet'.
install-package : Unable to install, multiple packages matched 'firefox'. Please
At line:1 char:1
+ install-package firefox
+ ~~~~~~
+ CategoryInfo          : InvalidArgument: (Microsoft.Power....InstallPackage) [Install-Package]
+ FullyQualifiedErrorId : DisambiguateForInstall,Microsoft.PowerShell.PackageManagement.Cmdlets.InstallPackage
```

PS C:\>

Želimo instalirati zadnju verziju Firefoxa iz Chocolatey galerije.

```
install-package -name "firefox" -providername "chocolatey"
```

Nakon minuti-dvije imat ćemo preglednik Firefox, baš kao na Linuxu. Kad Chocolatey objavi novu verziju Firefoxa, primijenit ćemo ju dodavanjem parametra -installupdate istoj naredbi. Isti princip vrijedi za sve druge pakete. Aplikacije možemo instalirati za sve korisnike računala ili za pojedinačnog, jasno, ovisno o zadanim parametrima. Prije prve instalacije neke aplikacije, mudro je pravovremeno se informirati o eventualnim ovisnostima (preduvjetima) jer o autoru paketa ovisi hoće li ih provjeravati.

Kako vidjeti osnovne podatke o aplikacijama instaliranim PacManom? Naredbom **get-package**. Dakako, tu je i naredba **uninstall-package** sa brojim opcijama, ističemo **-force** kojom eliminiramo eventualne upite, korisno za skriptni način primjene.

Do sada smo razglabali o upravljanju aplikacijama. Primjena zakrpa s Windows Update zasebna je priča. Moramo iz PSGallery povući namjenski modul **PSWindowsUpdate** i onda rabiti naredbe tog modula.

```
install-package -module pswindowsupdate
get-command -module pswindowsupdate
```

Svašta se još dade reći o PacManu, još više toga o infrastrukturi, po raznim aspektima, od funkcionalnog do sigurnosnog, ali bit će za to prilike. Događaje možemo pratiti kroz Event Viewer > Windows Logs > Applications and Services Logs > Microsoft > Windows > Powershell ili, ako nam je komandna linija baš po mjeri u svakoj situaciji, naredbom

```
get-winevent microsoft-windows-powershell/operational -filterxpath * (ili slično)
```

Na kraju jedna zanimljivost: Trenutno se PacMan može rabiti samo iz komandne linije, ali već se radi na API-u kojime će se omogućiti izrada GUI front-end-a! Nešto kao Apt i Aptitude, Nmap i Zenmap.... Beskrajno je inspirativna ta navada zagovaranja komandne linije kao prava alata za prava informatičare, a onda se traže načini kako se izvući iz nje! :o)

pon, 2016-07-25 14:03 - Ratko ŽižekVijesti: [Windows](#) [2]

Vote: 0

No votes yet

Source URL: <https://sysportal.carnet.hr/node/1666>

Links

- [1] <https://sysportal.carnet.hr/node/1641>
- [2] <https://sysportal.carnet.hr/taxonomy/term/12>