Windows Just Got Chocolatey (Package Management)



Rob Reynolds, Puppet Labs

usenix LISA15

November 8–13, 2015 | Washington, D.C.
www.usenix.org/lisa15 #lisa15

Rob Reynolds

- Developer at Puppet Labs
- Creator of Chocolatey
- Background in .NET development (ASP.NET MVC, JavaScript, C#, Windows services, MSMQ)
- Co-wrote and maintained infrastructure framework known as the Chuck Norris Framework
- Doing the hard work for Puppet on Windows so you don't have to



Agenda

- What is Chocolatey?
- Windows 10 Built-in Package Manager
- Building on Windows Ecosystem
- A history and the future of Chocolatey
- Configuration Management



Binary Package Manager for Windows

- Behaves much like Yum
- Uses NuGet packaging framework + PowerShell for automation scripts
- CLI tool - choco.exe
- GUI - ChocolateyGUI
- Allows non-centralized/private repositories
- Microsoft validated - there will be a OneGet provider





chocolatey.org (Community Repo)

- Community Feed / Community Maintained
- Moderated as of October 2014
- A couple people have downloaded packages (1 yr ago at 5 Million)

2,938 unique packages

33,144,406 total downloads

- Organizations should not depend on community b/c trust and control



But Windows 10 has a Package Manager Built-in!

Sort of…

OneGet aka PowerShell PackageManagement

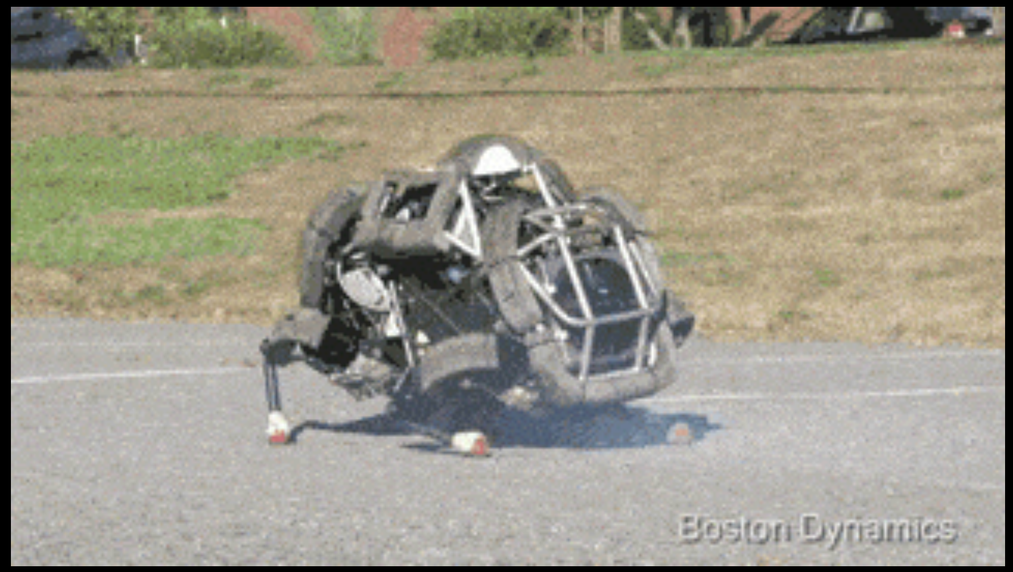
- Package Management may be a misnomer
- OneGet is ***NOT*** a package manager. It's a package manager *aggregator*.
- The PowerShell Provider ***is*** included.
- It knows how to bootstrap providers.
- Chocolatey will have a provider for OneGet.

Building on an Existing Windows Ecosystem

Windows Software Ecosystem

- State of The Union:
- Thousands of existing installers and archive files
- It’s like the ultimate Wild West for a packaging system





Unify the Windows Packaging Ecosystem?



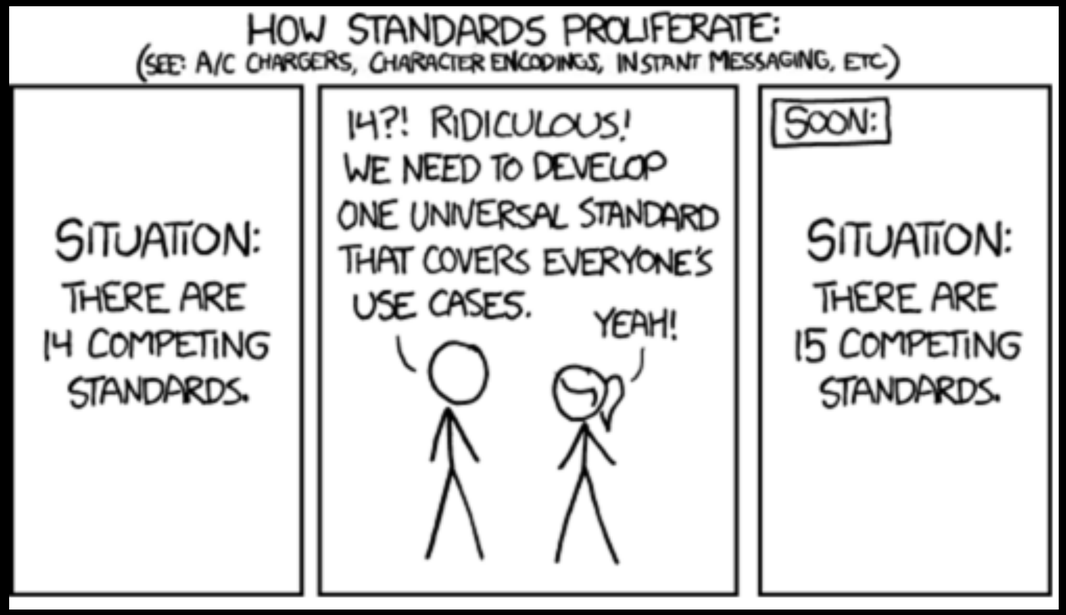
Unify the Windows Packaging Ecosystem?



Unify the Windows Packaging Ecosystem?

Unify the ecosystem

- You'd need something that abstracts (builds on top of) the existing ecosystem and provides a single unifying interface
- Failed package managers for Windows have left a trail attempting to ignore the above and build a new standard
- "Ain't nobody got time for that!"



Keep it Simple S___

How not to KISS

- Make it complicated
- Unnecessarily complex
- Over-engineer
- Magpie



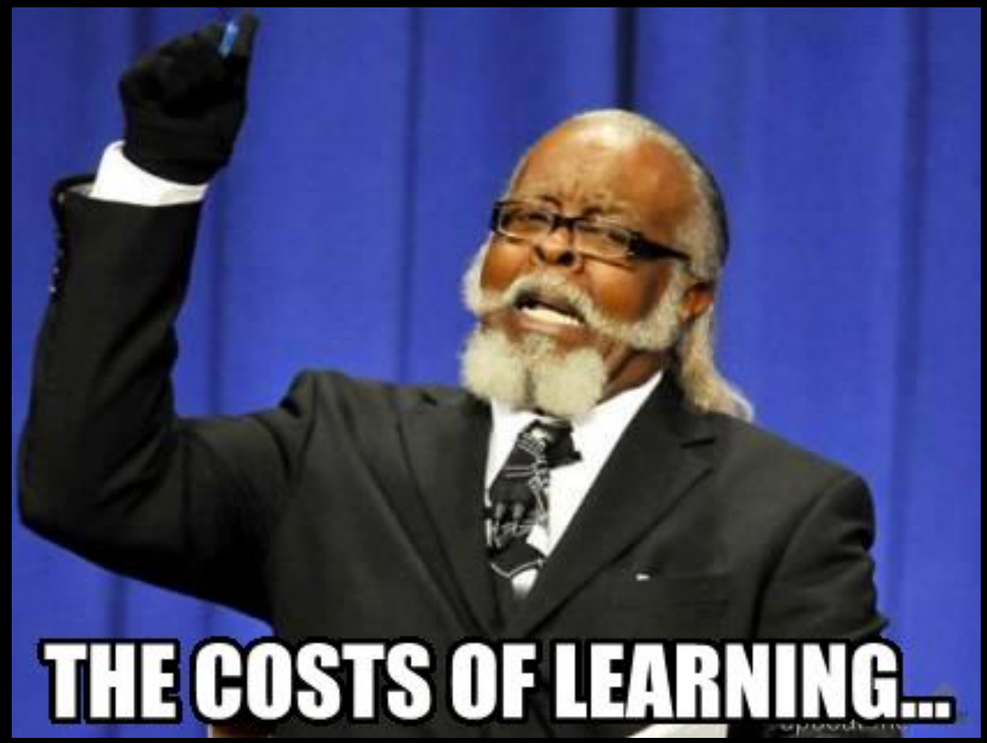
How to KISS

- “Simple ain’t easy” - Thelonious Monk
- Complex != Complicated
- Introduce Complexity only when necessary

- Resist the urge to magpie

Building a Community

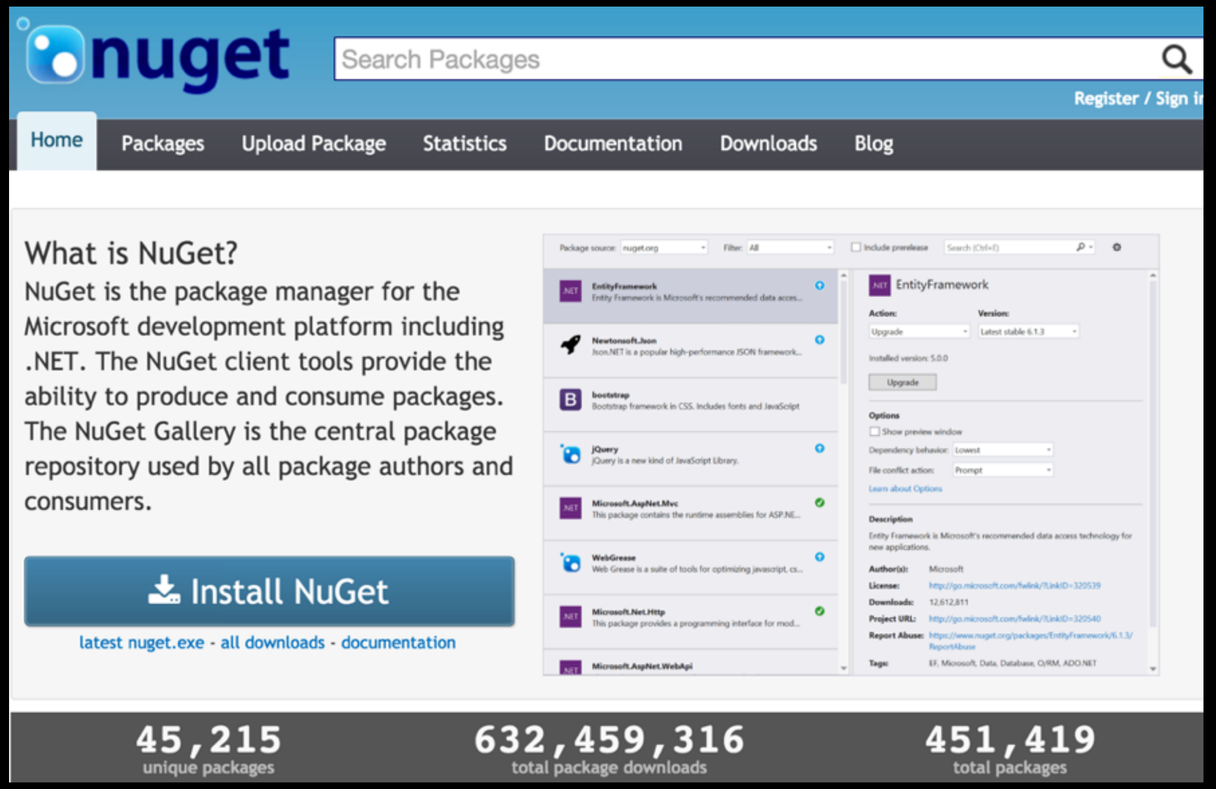
- Foster friendliness and a helpful attitude
- Document, document, document
- Business friendly
- Open Source
- Community Package Repository
- Make the UX Simple
- Did I mention document? Because reasons.



The NuGet Packaging Framework

NuGet - 2010

- NuGet is the package manager for the Microsoft development platform including .NET.
- I was on the team that brought about the “Nu” part. We started working on the NuGet team in 2010.
- Not the first FOSS project for Microsoft (WiX and Orchard), but perhaps one with the biggest contributions
- Looking back it’s completely different now with everything Microsoft being open-sourced and they are really trying not to be evil.



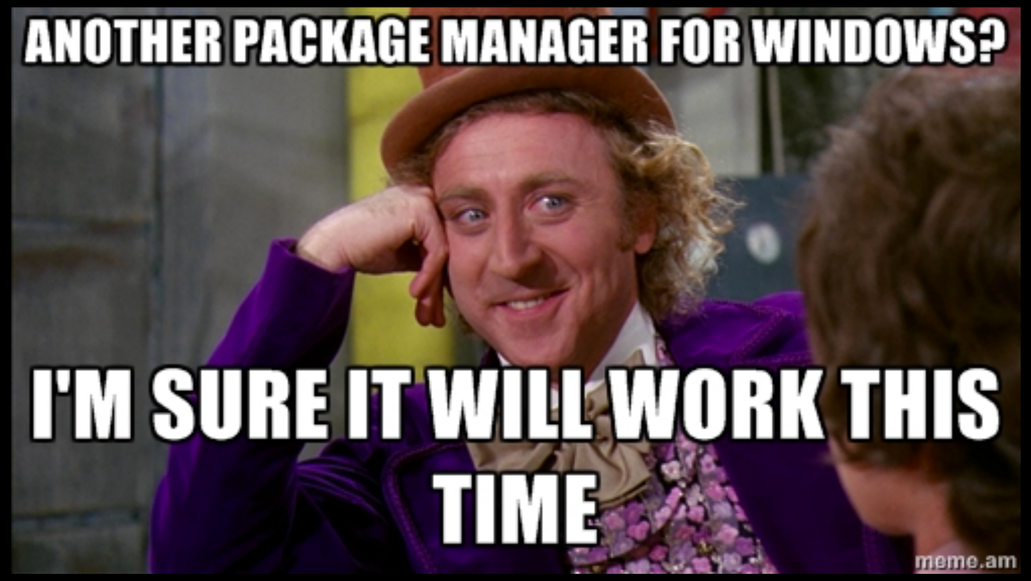
NuGet as a Framework

- Packaging
- Decentralized
- Describe dependencies
  - Version ranges
- No conflicts, provides
- Not architecture specific
- No signing

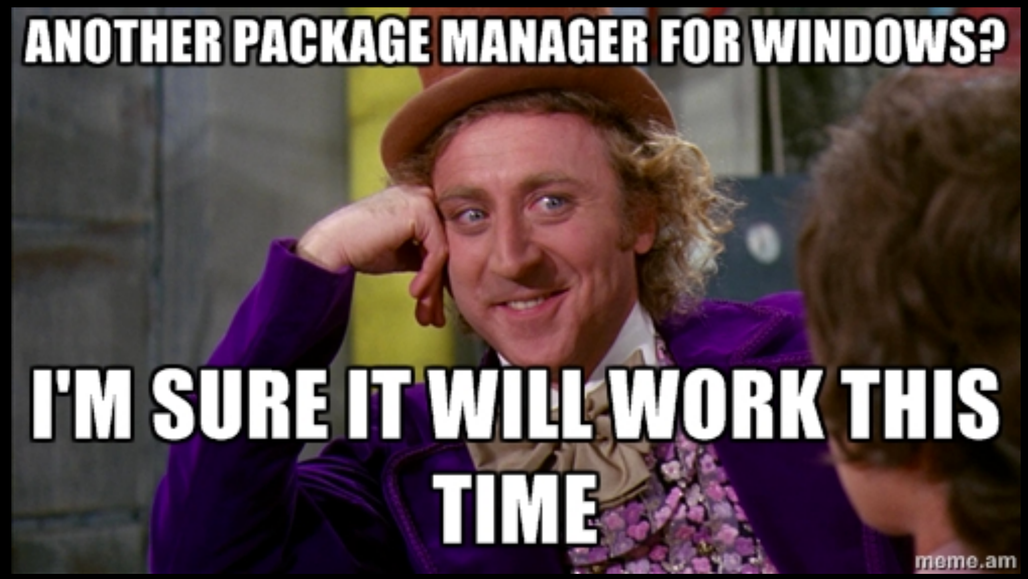
windirstat.nuspec

```
<?xml version="1.0"?>
<package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:
xsd="http://www.w3.org/2001/XMLSchema">
  <metadata>
    <id>windirstat</id>
    <title>WinDirStat</title>
    <version>1.1.2.20140318</version>
    <authors>Oliver Schneider</authors>
    <owners>Rob Reynolds</owners>
    <summary>WinDirStat - Windows Directory Statistics</summary>
    <description>WinDirStat is a disk usage statistics viewer and
    cleanup tool for Microsoft Windows.</description>
    <projectUrl>http://windirstat.info</projectUrl>
    <iconUrl>https://raw.githubusercontent.com/ferventcoder/chocolatey
    -packages/master/icons/windirstat.png</iconUrl>
    <tags>windirstat disk usage admin</tags>
    <licenseUrl>http://www.gnu.org/licenses/gpl-2.0.html</licenseUrl>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
  </metadata>
  <files>
    <file src="tools/**" target="tools" />
  </files>
</package>
```

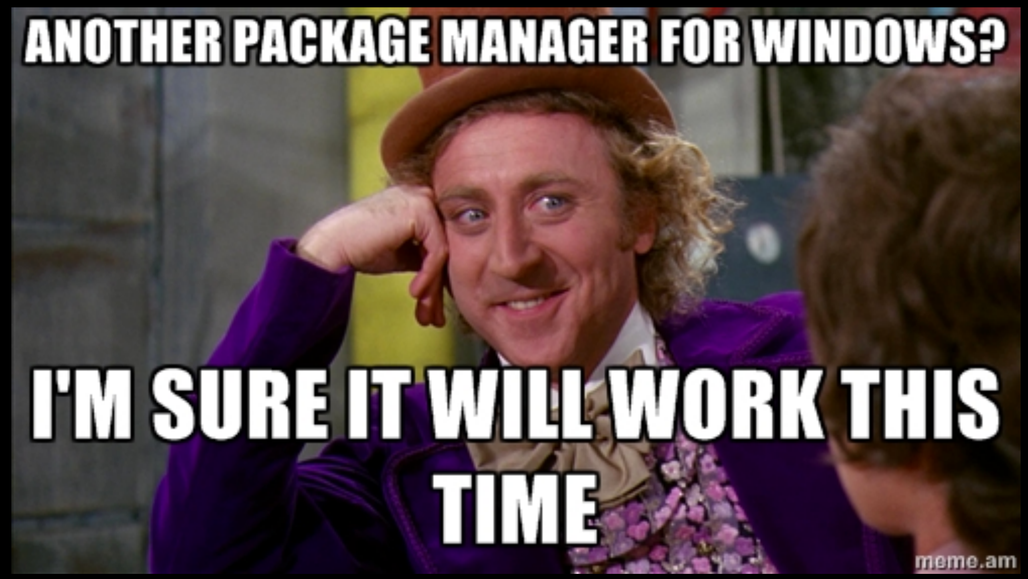
Owners == package maintainers



Tell Me More About Chocolatey



It started as a thought experiment.



Seriously, did you ever look at the source?!

What I Wanted in 2011

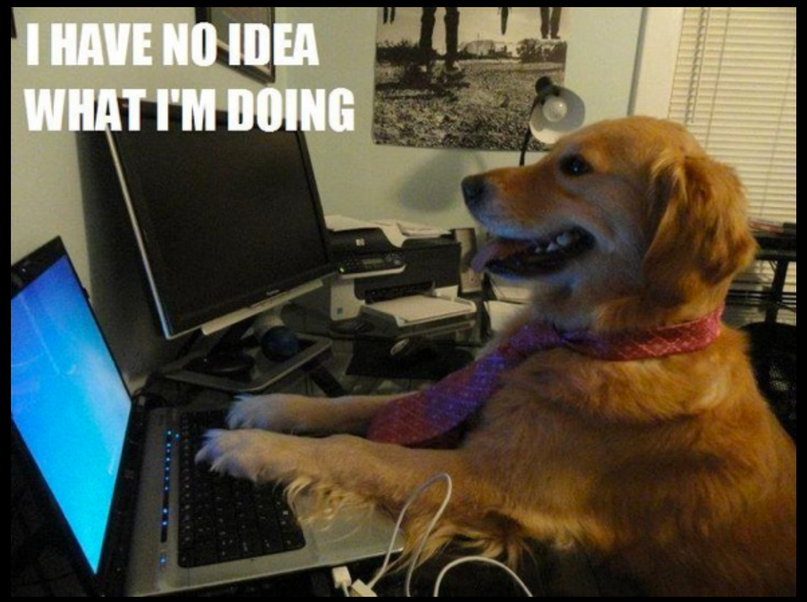
- Pairing / troubleshooting experience - No Notepad++?
- Experience with silent installers
- Existing?
- How to globalize?

Experiment in March 2011

- Building onto NuGet
- Packages tagged “chocolatey” on nuget.org (until Sep 2011)
- Concepts of KISS packaging
- Building on an existing ecosystem
- Focus on UX

The Experiment Was Built In PowerShell

- PowerShell
- Built with PowerShell as a programming language?!
- Always felt like I was relearning PowerShell





And Then… Community

- Somewhere along the line came the community
- Design or Sheer Luck?
- Maybe it was the semi-okay Documentation?
- 1. Gnomes. 2. ??? 3. Profit!

Automatic packages

- Built with Ketarin to check for updates and push up newer versions of packages automatically
- Some folks have taken advantage of that



Microsoft Got Involved

- Validated the idea in 2014
- Announced OneGet
- Talked about integration
- The OneGet boot strapper will bring Chocolatey to Windows seamlessly the first time you ask for it, as if it was built-in.

Ensuring Chocolatey for the Long Term

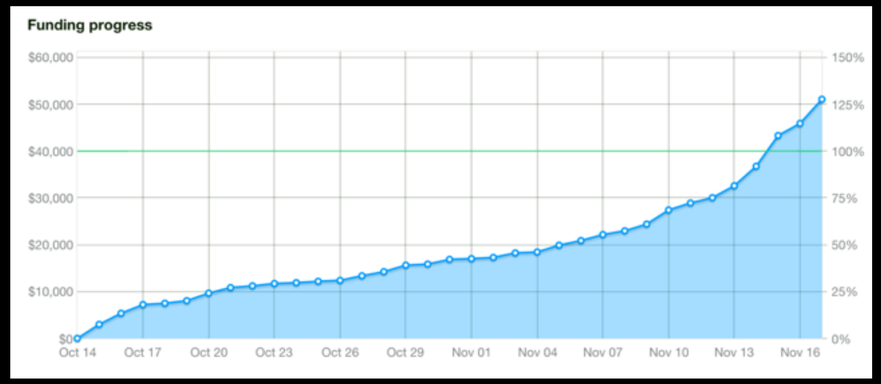
The Long Overdue Rewrite - 2014

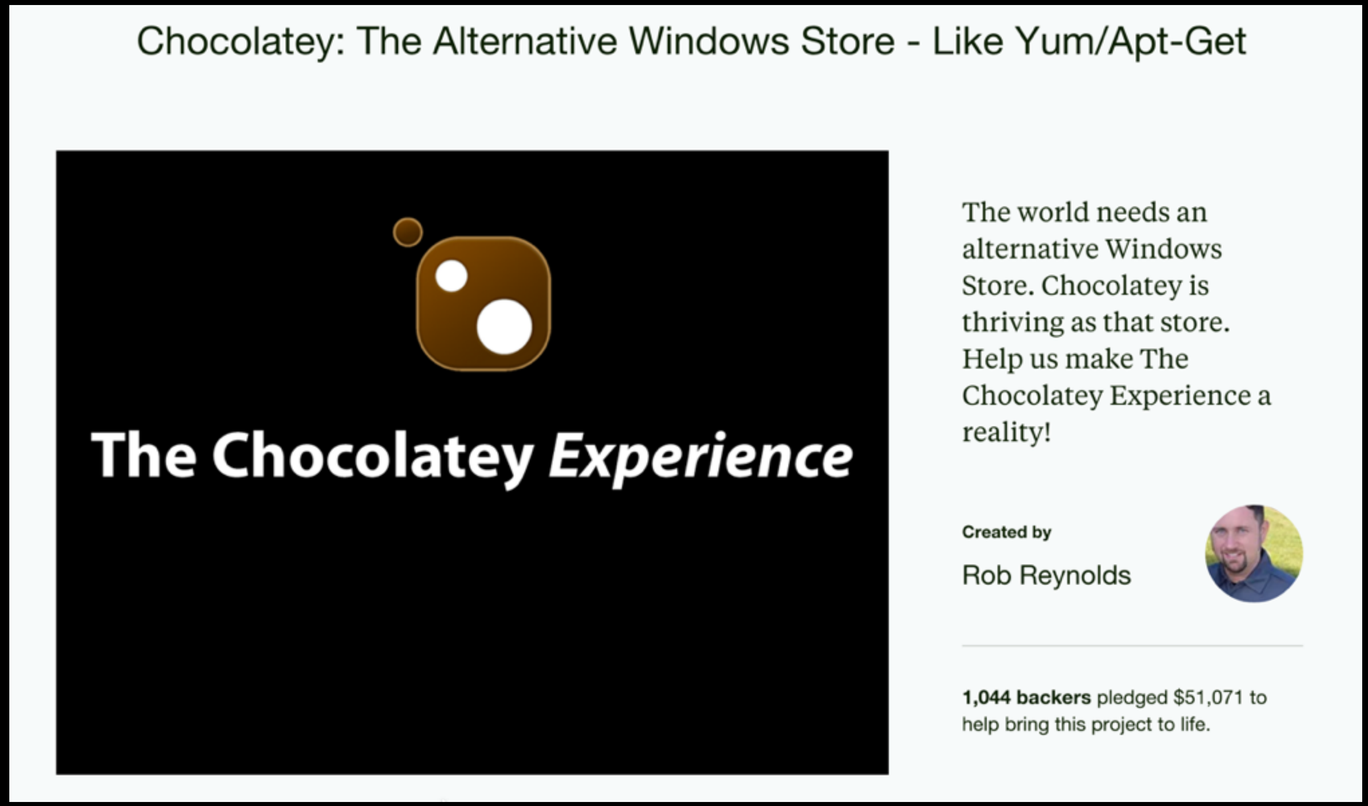
- Started March 2014, first release in 0.9.9 in March 2015.
- C# - b/c already wrote production code while learning language
- Why?
  - Maintainability
  - Speed
  - Better library support
  - Strong techniques for security
  - Because pro/business



Kickstarter - October 2014

- Started planning in June
- Introduces the ideas of Chocolatey for Business
- Pro/Business Version to offer additional features
- Longevity of Chocolatey itself



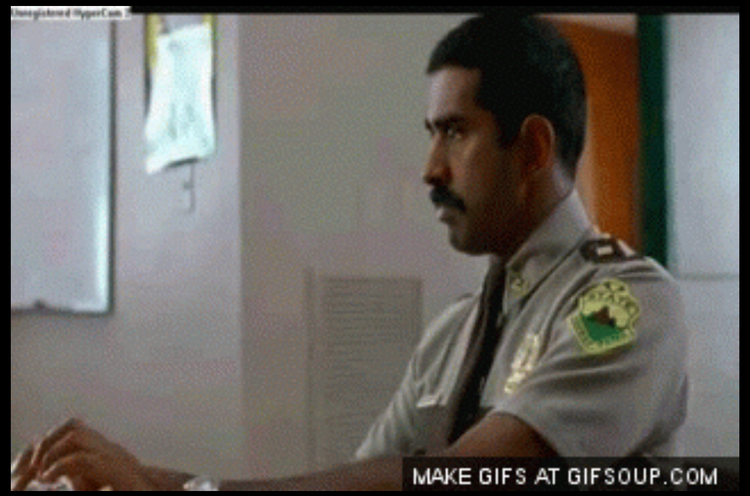


Moderation of Community Packages

- Introduced for more predictability
- Package quality
- Safety and Security
- Starting to build trust in community repository, but still quite a long way to go

Extending NuGet

- Enhanced nuspec with provides, conflicts, replaces (although choco doesn't know what to do with those yet)
- Will be enhancing further: architecture, os versions, etc
  - NVRA (name version release architecture)
- Package signing (PGP) / Security Enhancements



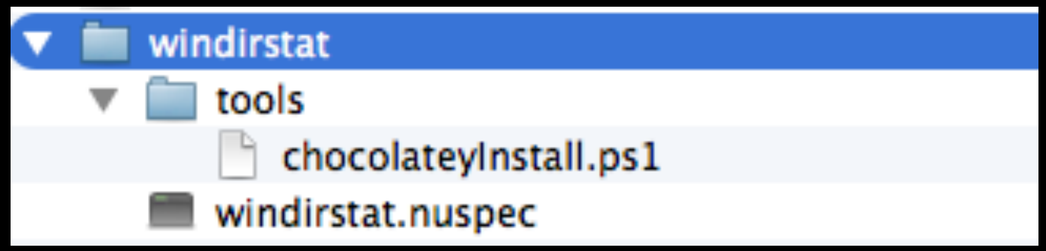
More Future Enhancements

- More security and traceability
- Try to maintain good balance of KISS while introducing necessary complexity
- Chocolatey for Business / Enterprise General Availability

Using Chocolatey

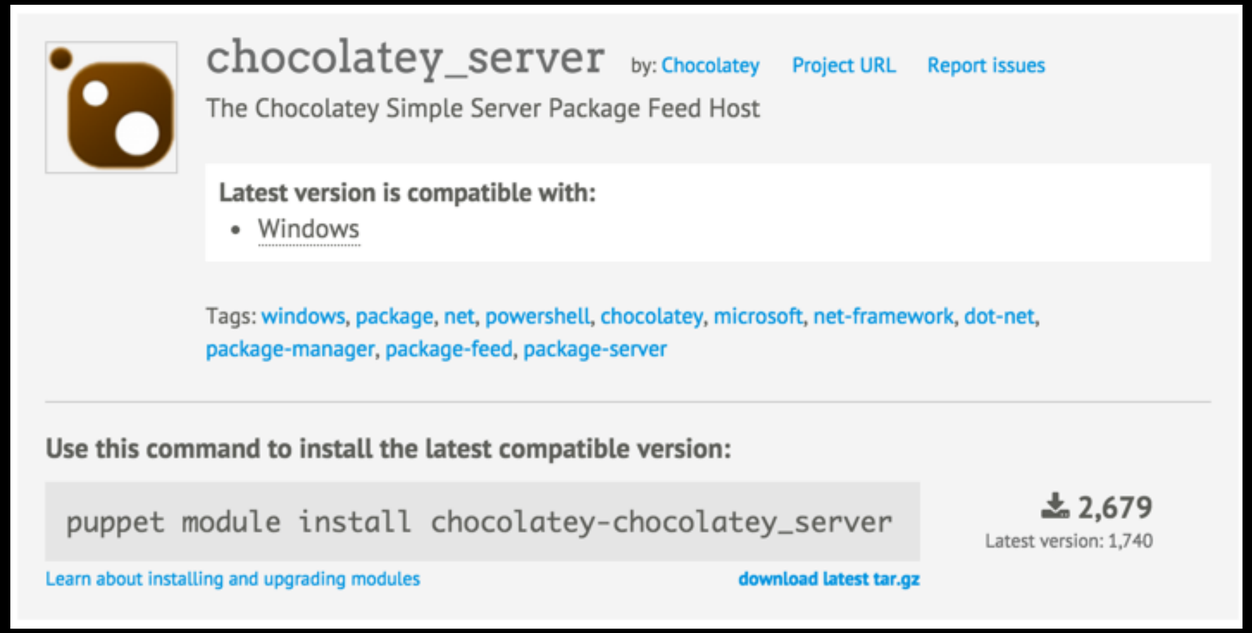
Creating Packages

- choco new -h
- Create organizational templates!
- *.nuspec
- chocolateyInstall.ps1
- Other resources like binaries / config files, etc
- Getting started / reference at http://bit.ly/choco-create



Hosting Your Own Packages

- With Puppet - chocolatey/ chocolatey_server module
- Non-Windows Hosting
  - Artifactory
  - Sonatype Nexus
- NuGet Gallery proper
- ProGet / MyGet
- https://bit.ly/choco-package-server



Configuration Management

CM Integrations

- Puppet Labs - https://bit.ly/choco_puppet (Approved, soon Supported)
- Chef - https://bit.ly/choco_chef
- Ansible - https://bit.ly/choco_ansible
- Saltstack - https://bit.ly/choco_salt
- PowerShell DSC - https://bit.ly/choco_dsc



Demo

Questions?

- Join the newsletter for announcements - https://chocolatey.org/#newsletter
- ferventcoder { twitter, github, gmail}