



Chocolatey

The package manager for Windows

Chocolatey + Artifactory = A Sweet Solution For Managing Windows



Rob Reynolds

Founder

Chocolatey Software

@ferventcoder



JFrog Artifactory



Baruch Sadogursky

JFrog Developer Advocate

JFrog

@jbaruch





Rob Reynolds

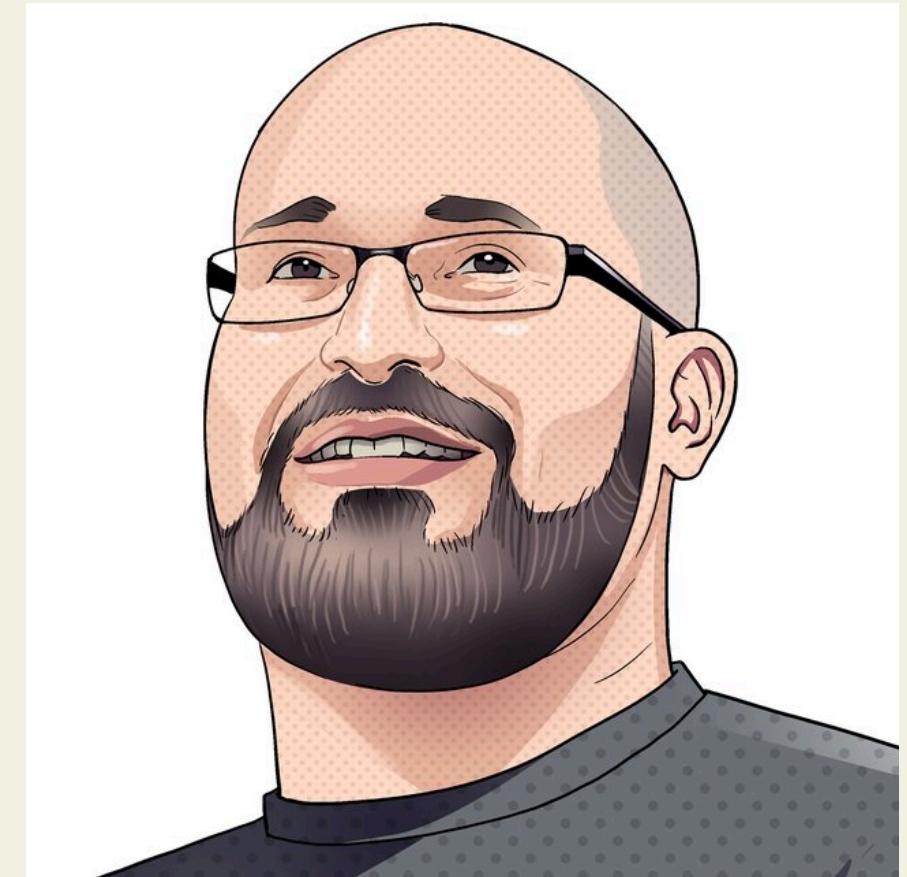
- Creator and Founder of Chocolatey 
- Enjoys long walks on the beach and designing solutions that make hard things easy
- Co-wrote infrastructure framework known as the Chuck Norris Framework
- Over 10 years experience in infrastructure automation
- Obsesses over user experience
- Microsoft MVP for whatever they are calling .NET area now





Baruch Sadogursky

- JFrog Developer Advocate 
- Can speak about literally any topic (such as My Little Pony), prefers all things tech
- a.k.a. JBaruch / @jbaruch
- Over 10 years experience in repository management with Artifactory and Bintray
- International speaker - presents on Continuous Delivery, Java, DevOps, and of course Artifactory among other topics





Agenda

- Overview of JFrog Artifactory
- Overview of Chocolatey
- Integration and Use Case
- Demo and Q&A
- Questions





Artifactory





JFrog Artifactory



- Artifact repository
- Can manage almost any type of artifact





Artifactory Features

- Reliability
- Efficiency
- Security
- Stability
- Productivity
- XPlat Installation (Windows or other)





TeamCity



Hudson



drone.io

wercker



Travis CI

circleci



CODESHIP



JFrog Artifactory



Remote Repositories





Artifactory Dashboard

The screenshot shows the JFrog Artifactory web interface. At the top, there's a navigation bar with a user icon, a search icon, and a help icon. The main title is "JFrog Artifactory". Below the title, a large message says "Artifactory is happily serving 0 artifacts". It also displays "Artifactory Version 5.5.1 (Uptime is 0d 0h 14m 0s)" and "Latest Release: 5.5.1". On the left, there's a sidebar with icons for Home, Artifacts, Properties, Checksums, and Help. Below the sidebar are two rows of links: "Quick Search" (Package, Archive, Property, Checksum, JCenter Search, Search), "User Guide", "Webinar Signup", "Support Portal", "Stackoverflow", "Blog", and "Rest API". In the center, there are three main sections: "Set Me Up" (with a "chocolatey-local" repository key), "Last Deployed Builds" (which says "No builds to display. Learn how to integrate your build information with Artifactory."), and "Most Downloaded Artifacts" (which says "No artifacts to display. Learn how to deploy your artifacts to Artifactory."). At the bottom, there are links for "High Availability", "JFrog Bintray Distribution", "JFrog Xray", "JFrog CLI", "Build Integration", "Activate Windows", "Go to System in Control Panel to activate Windows.", and "Docker". A note at the bottom states: "Other than JFrog's trademarks, marks and logos, all other trademarks displayed in this application are owned by their respective holders. JFrog is not sponsored by, endorsed by or affiliated with the holders of these trademarks. ...".





Why Chocolatey?





**Software management may
account for 50-90% of what
you do on Windows**



Windows Software Ecosystem

- Over 20 installer formats and thousands of installers in the wild
- Zips and other archive formats
- Software installers are messy
- Deploy internal/3rd party software using same approach?
- It's like the Wild West





Why Chocolatey?

- Reporting
- Features
- Offline
- Secure
- Flexibility
- Beyond Programs and Features
- Beyond just running installers
- Integrates with everything

Will Button
@wfbutton

Follow

Created my first @chocolateynuget package. It was a pretty #awesome experience.

RETWEETS LIKES

3 4

11:26 PM - 31 May 2016

3 4





Why Chocolatey?

- Focused solution - does 1 thing very well for over 6+ years
- Uses PowerShell - not learning new or proprietary technologies
- One approach for all software management - 3rd party and internal
- Reinventing Chocolatey could cost over \$1 million
- Hundreds of other companies are using it, there must be a reason





What is Chocolatey?





What is Chocolatey?

- Software Management
- PowerShell
- Fancy zip files = “packages”
- Packages are independent software deployment artifacts
- Takes your deployment scripts to the next level





Chocolatey - Sane Software Management

- Easily manage the software lifecycle
- Native installers, zips, binaries
- PowerShell module simplifies work
- Packages are independent building blocks
- Integrates with configuration management and RMM tools
- Business friendly features





Chocolatey - The Approach

6+ years - proven technology

- Single, unifying interface - control the chaos
- Decentralized - multiple places to get packages, including internal
- PowerShell
- Flexible
- Secure
- Reliable





Sweet Features





FOSS vs Chocolatey for Business (C4B)



- **Chocolatey open source (FOSS) == package management**
 - Works well in organizational use
- **C4B == complete software management**
 - Smoother experience
 - Builds on top of FOSS
 - Better system integration, endpoint management, etc
 - Better visual interfaces (GUIs)
 - Features geared specifically towards organizational use



Smart Installs / Smarter Upgrades

```
Administrator: Windows PowerShell
PS C:\packages> choco install git.install --source . -y
Chocolatey v0.9.10-beta1-426-g3c8b16e Business
Installing the following packages:
git.install
By installing you accept licenses for the packages.

git.install v2.8.4
Loading 'chocolatey.licensed' extension.
Copying git.install
  from 'C:\ProgramData\chocolatey\lib\git.install\tools\files\Git-2.8.4-64-bit.exe'
Virus checking skipped/turned off. Not verifying 'Git-2.8.4-64-bit.exe'
Installing Git-2.8.4-64-bit.exe...

Git-2.8.4-64-bit.exe has been installed.
WARNING: Git installed - You may need to close and reopen your shell for PATH changes to take effect.
Environment Vars (like PATH) have changed. Close/reopen your shell to see the changes (or in powershell/cmd.exe just type `refreshenv`).
The install of git.install was successful.
Software installed to 'C:\Program Files\Git\' ←

Chocolatey installed 1/1 package(s). 0 package(s) failed.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\packages> git
git : The term 'git' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ git
+ ~~~
    + CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFoundException
    + FullyQualifiedErrorId : CommandNotFoundException

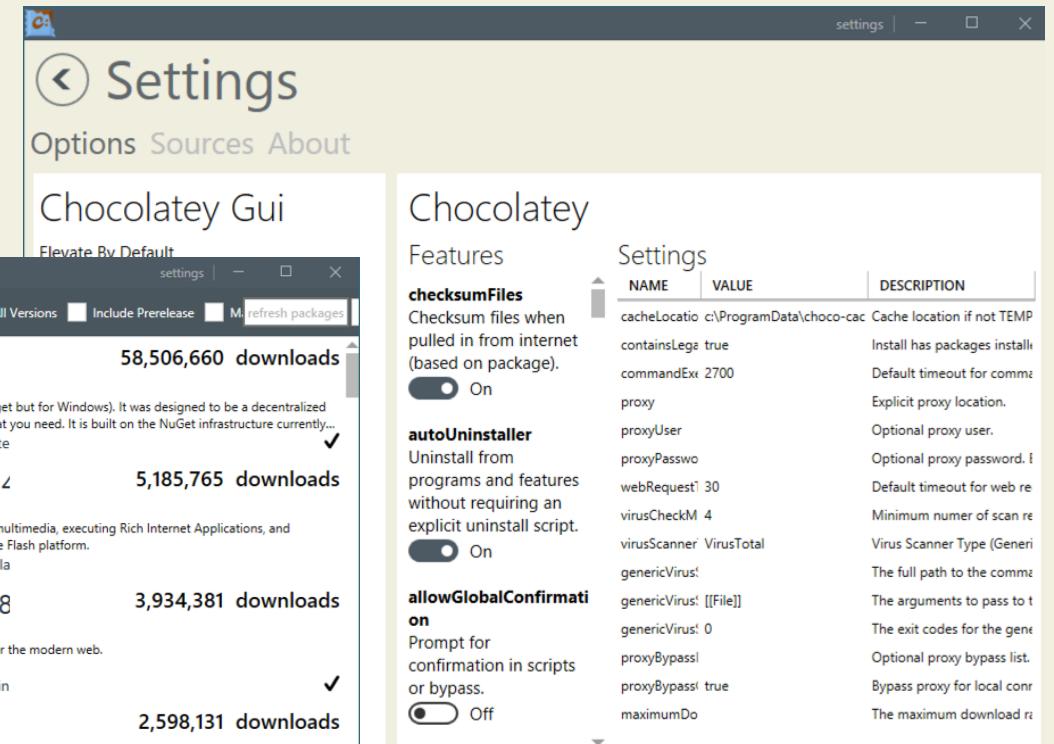
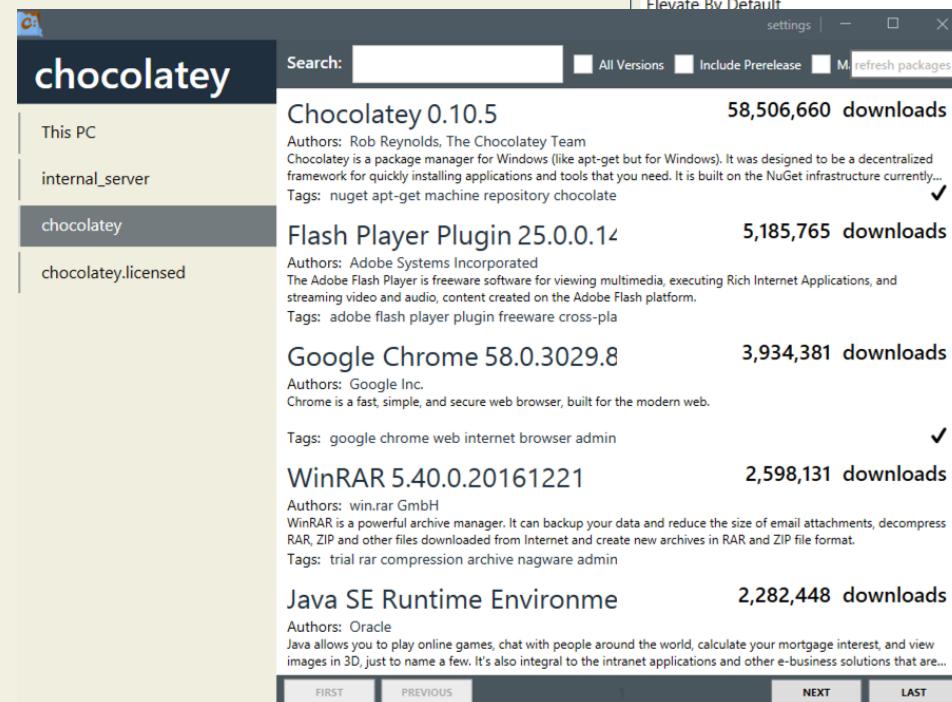
PS C:\packages> refreshenv ←
Refreshing environment variables from the registry...
Finished
PS C:\packages> git
usage: git [--version] [--help] [-c <path>] [-c name=value]
  [--exec-path=<path>] [--html-path] [--man-path] [--info-path]
```

- Track software installation in Programs & Features to map packages to system installs
- Track environment variable changes
- Leave out guesswork by reporting on software installation location
- Upgrades - Use XML Document Transform (XDT) for .NET configuration file changes
- Exclusive to Chocolatey - Use new tools / commands without closing / re-opening your shell for environment changes to take effect



Graphical Interface - Chocolatey GUI

- Configure Chocolatey
- Manage software installations





Automatic Uninstallation

```
PS C:\packages> choco uninstall 1password -y
Chocolatey v0.9.10-beta-422-g335bc01 Business
Uninstalling the following packages:
1password

1password v4.6.0.598
Running auto uninstaller...
Auto uninstaller has successfully uninstalled 1password.
1password has been successfully uninstalled.

Chocolatey uninstalled 1/1 packages. 0 packages failed.
See the log for details (C:\ProgramData\chocolatey\logs\choco.log)
```

- Exclusive to Chocolatey - Automatically uninstall over 90% of installations without uninstall script
- AutoUninstaller reduces time by not developing and managing uninstall scripts





Integrates with Everything

- <https://chocolatey.org/docs/features-infrastructure-automation>
- Puppet
- Chef (built-in), cookbook
- PowerShell DSC
- Ansible
- Saltstack
- Octopus Deploy
- Boxstarter
- SCCM
- RMM, or really anything that can run scripts





Reporting / Auditing





Inventory - Comprehensive Software Audit

```
PS C:\> choco list -lo --include-programs
1password 4.6.0.603
chocolatey 0.9.10.3
chocolatey.extension 1.4.0
chocolatey.server 0.1.1
DotNet4.0 4.0.30319.20141222
git 2.7.0.0
Googlechrome 51.0.2704.106
roundhouse 0.8.5.0
8 packages installed.

7-zip 9.22 (x64 edition)|9.22.00.0
Bitvise SSH Server 5.60 (remove only)|
ChocolateyGUI 0.13.2.0|0.13.2.0
Launchy 2.5|
Notepad++|6.4.2
Oracle VM VirtualBox Guest Additions 4.3.12|4.3.12.0
Puppet (64-bit)|3.8.3
7 applications not managed with chocolatey.
```

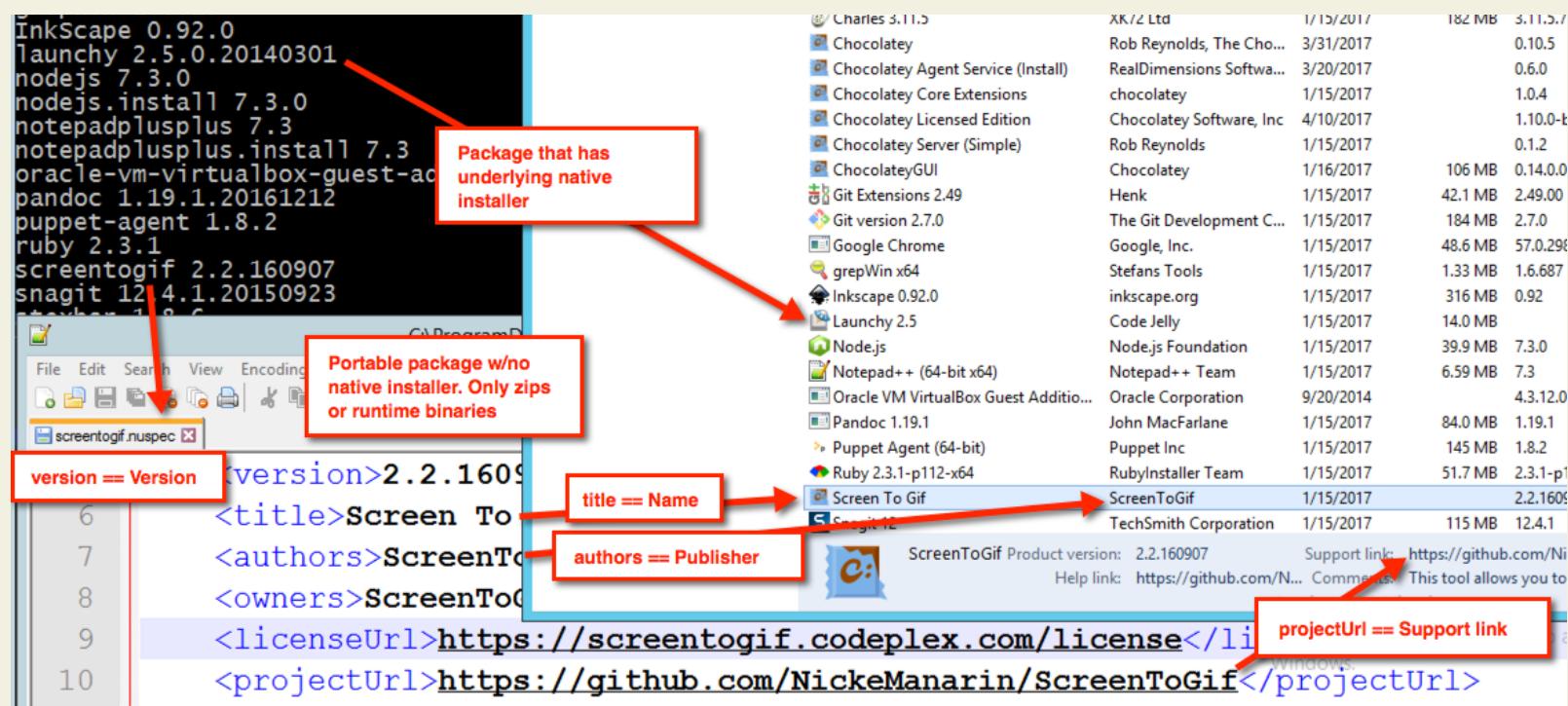
- Exclusive to Chocolatey - Visibility beyond just the software 'installed' in Programs & Features
- Track versions of zip archives & own internal software that is not visible in system installation.
- Get a complete view of your systems. Chocolatey reports applications in systems not directly under its management.
- Package Audit (C4B) adds who and when

```
PS C:\packages\createpackages> choco list -lo --audit -r
1password|4.6.0.598|User:vagrant|Domain:WIN2012R2X64|Original User:vagrant|Install Date:2017-07-09 16:53:17Z
7zip|16.02.0.20160811
baretail|3.50.0.20120226
```



Package Synchronizer (C4B) - All Packages in Programs and Features

- Programs and Features represents only 50-80% of the software on a machine
- Chocolatey brings managed entries for non-installer packages
- Supporting legacy inventory reporting systems is now a snap
- Do not need to build MSIs / Installers for internal use just to support legacy reporting





Logging - Detailed Reporting

The screenshot shows a terminal window with the following details:

- Tab bar: Open, Highlighting, Follow Tail, ANSI, C:\ProgramData\chocolatey\logs\chocolatey.log (5.9 MB)
- Log content:

```
2017-06-09 21:55:52,794 2236 [DEBUG] - Configured chocolatey.infrastructure.logging.log4net.config.xml from assembly choco, Version=0.10.7.0, Culture=neutral, PublicKeyToken=8d4d09c1f8e300a0, processorArchitecture=Unknown
2017-06-09 21:55:52,849 2236 [DEBUG] - XmlConfiguration is now operational
2017-06-09 21:55:52,857 2236 [DEBUG] - Evaluating license file found at 'C:\ProgramData\chocolatey\license\chocolatey.license'.
2017-06-09 21:55:53,044 2236 [DEBUG] - Loading Licensed Extensions and Commands
2017-06-09 21:55:53,083 2236 [INFO ] - =====
2017-06-09 21:55:53,411 2236 [INFO ] - Chocolatey v0.10.7 Business
2017-06-09 21:55:53,427 2236 [DEBUG] - Chocolatey is running on Windows v 6.3.9600.0
2017-06-09 21:55:53,427 2236 [DEBUG] - Attempting to delete file "C:\ProgramData\chocolatey\choco.exe.old".
2017-06-09 21:55:53,435 2236 [DEBUG] - Attempting to delete file "C:\ProgramData\chocolatey\choco.exe.old".
2017-06-09 21:55:53,443 2236 [DEBUG] - Command line: "C:\ProgramData\chocolatey\choco.exe" upgrade chocolatey.extension -s C:\vagrant\resources\packages -y -f
2017-06-09 21:55:53,443 2236 [DEBUG] - Received arguments: upgrade chocolatey.extension -s C:\vagrant\resources\packages -y -f
2017-06-09 21:55:53,536 2236 [DEBUG] - RemovePendingPackagesTask is now ready and waiting for PreRunMessage.
2017-06-09 21:55:53,544 2236 [DEBUG] - CountdownTask is now ready and waiting for PostRunMessage.
2017-06-09 21:55:53,544 2236 [DEBUG] - Terminal services setup not necessary for this session.
2017-06-09 21:55:53,552 2236 [DEBUG] - SynchronizeTask is now ready and waiting for PreRunMessage.
2017-06-09 21:55:53,552 2236 [DEBUG] - PackagesInProgramsAndFeaturesTask is now ready and waiting for PostRunMessage.
2017-06-09 21:55:53,560 2236 [DEBUG] - Sending message 'PreRunMessage' out if there are subscribers...
2017-06-09 21:55:53,575 2236 [DEBUG] - [Pending] Removing all pending packages that should not be considered installed...
2017-06-09 21:55:53,599 2236 [DEBUG] - [Synchronize] Updating packages with Programs and Features
2017-06-09 21:55:53,599 2236 [DEBUG] - [Synchronize] Currently only supports removals. Soon will handle updates.
2017-06-09 21:55:53,872 2236 [DEBUG] - 7zip 16.02.0.20160811
2017-06-09 21:55:53,880 2236 [DEBUG] - baretail 3.50.0.20120226
2017-06-09 21:55:53,950 2236 [DEBUG] - Charles 3.11.5
2017-06-09 21:55:53,950 2236 [DEBUG] - chocolatey 0.10.7
2017-06-09 21:55:53,950 2236 [DEBUG] - chocolatey-agent 0.6.0
2017-06-09 21:55:53,950 2236 [DEBUG] - chocolatey-core.extension 1.0.4
2017-06-09 21:55:53,950 2236 [DEBUG] - chocolatey.extension 1.10.0
2017-06-09 21:55:53,958 2236 [DEBUG] - chocolatey.server 0.1.2
2017-06-09 21:55:53,958 2236 [DEBUG] - ChocolateyGUI 0.14.0-unstable0165
2017-06-09 21:55:53,958 2236 [DEBUG] - DotNet4.0 4.0.30319.20141222
2017-06-09 21:55:53,958 2236 [DEBUG] - DotNet4.5.2 4.5.2.20140902
2017-06-09 21:55:53,958 2236 [DEBUG] - git 2.7.0.0
2017-06-09 21:55:53,958 2236 [DEBUG] - gitextensions 2.49
2017-06-09 21:55:53,958 2236 [DEBUG] - GoogleChrome 55.0.2883.87
2017-06-09 21:55:53,966 2236 [DEBUG] - grepwin 1.6.16
2017-06-09 21:55:53,966 2236 [DEBUG] - InkScape 0.92.0
2017-06-09 21:55:53,966 2236 [DEBUG] - l10n-en-US 0.5.0.20140901
```

- Log file captures very detailed information - environment and state
- Reduces rollback / recreation necessity



Compliance – Audit Out of Date Software

- choco outdated
- Comprehensive reporting on available upgrades
- See what will upgrade on next maintenance window

```
PS C:\packages> choco outdated
Chocolatey v0.9.10-beta1-426-g3c8b16e Business
Outdated Packages
  Output is package name | current version | available version | status

nodejs|4.4.5|6.2.1|false
nodejs.install|4.4.5|6.2.1|false
notepadplusplus|6.8.1|6.9.2|false
notepadplusplus.install|6.8.1|6.9.2|false
roundhouse|0.8.5.0|0.8.6|false

Chocolatey has determined 5 package(s) are outdated.
PS C:\packages>
```



Creating Software Deployment “Packages”





Chocolatey Packages

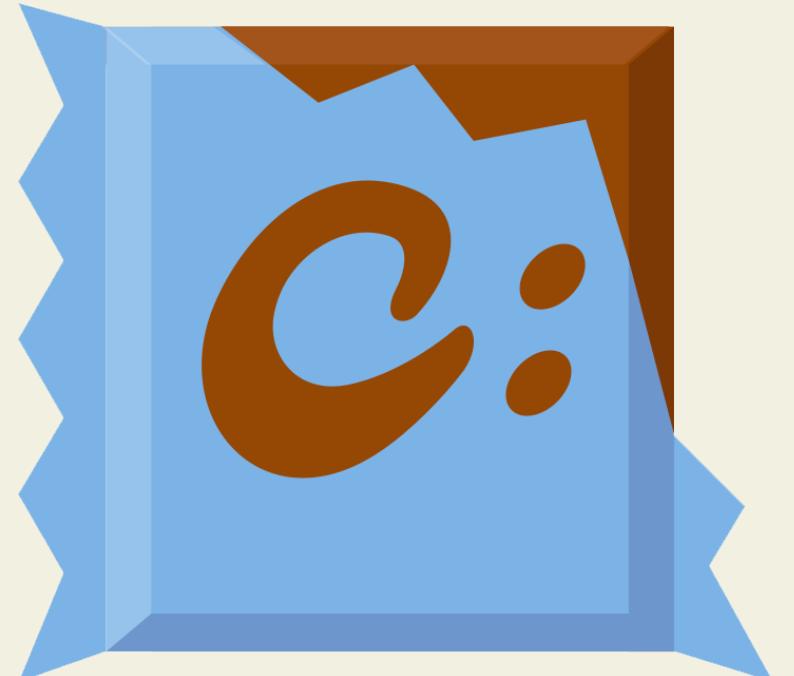
- Terminology: "Package" is nupkg file, "Software" is binaries or installers
- Zip files + Versioning, Dependencies & PowerShell
- Metadata - Software Information / Package Information
- PowerShell - <https://chocolatey.org/docs/helpers-reference>
 - 30+ Built-in functions - complex tasks become 1 liners
 - Handle additional missing installer logic (like add to PATH)
 - Add options missing from installer arguments
- Files - drop in binaries, native installers



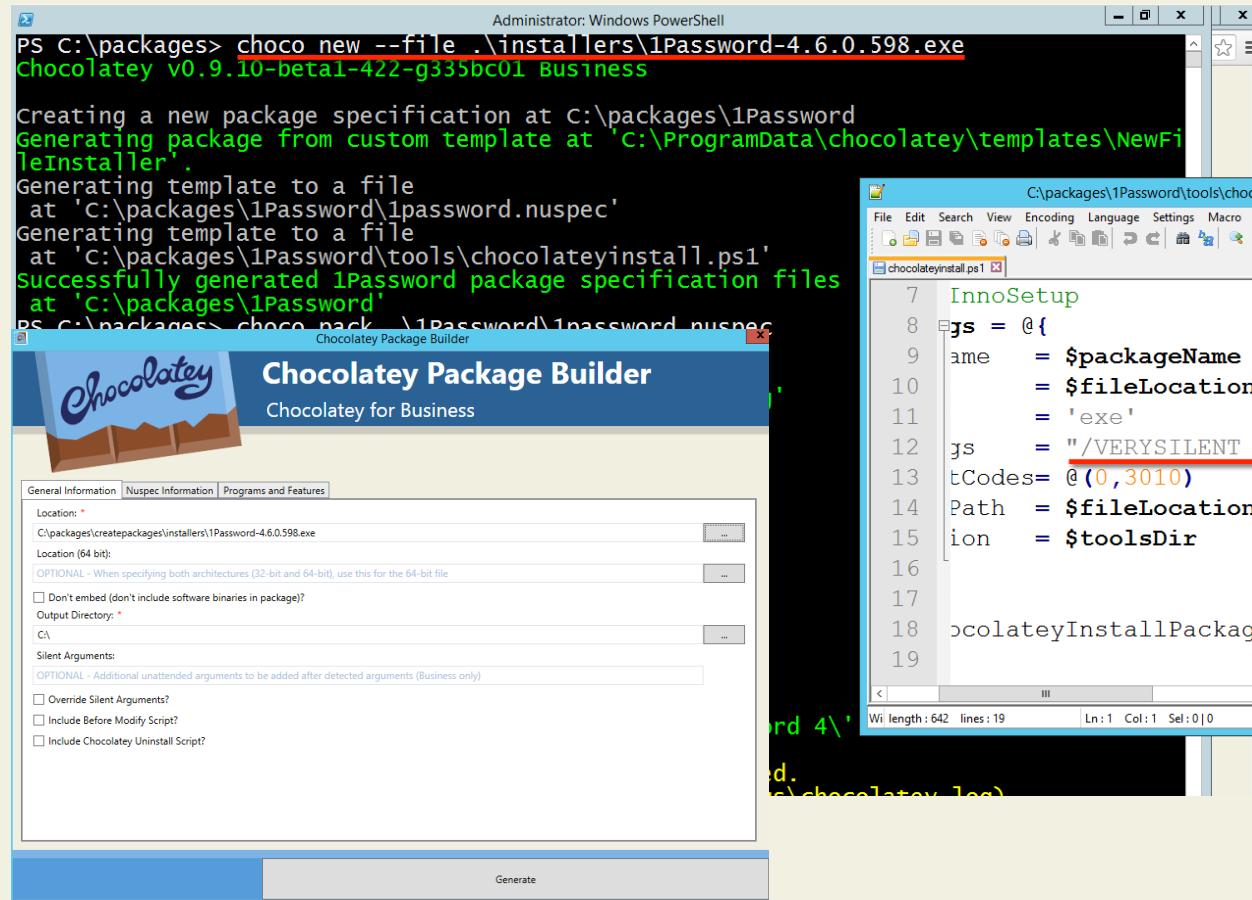


Creating Packages

- <https://chocolatey.org/docs/create-packages>
- choco new
 - Generates package files with most of the work done
 - Create your own packaging templates!



Package Builder (C4B) - “Generate software deployment packages in seconds”



- Package Builder automatically creates high quality packaging from an installer/zip in 5-10 seconds
- Determines everything necessary for silently installing software
- Supports passing urls (and checksums) - downloads files first
- Allows customization
- Supports exe, msi, 7z, zip, msu, and msp files



Package Internalizer (C4B) - “Convert existing packages to 100% offline and reliable”

- Take advantage of existing packages without dependency on the internet
- Downloads existing package and all remote resources
- Recompiles package to use those internal resources
- Option to download and point to other locations
- Ability to host packages on private internal repositories
- Visual Interface coming

```
PS C:\packages> choco download --recompile git.install
Chocolatey v0.9.10-beta1-426-g3c8b16e Business
Downloading an existing package to C:\packages\download

git.install v2.8.4 [Approved]
Found Install-ChocolateyPackage. Inspecting values for remote resources.
Found 'https://github.com/git-for-windows/git/releases/download/v2.8.4.windows-2.8.4-32-bit.exe' for replacement.
Found 'https://github.com/git-for-windows/git/releases/download/v2.8.4.windows-2.8.4-64-bit.exe' for replacement.

Download of 'Git-2.8.4-32-bit.exe' completed.
Replacing 'https://github.com/git-for-windows/git/releases/download/v2.8.4.windows-2.8.4-32-bit.exe' with '$(Split-Path -parent $MyInvocation.MyCommand.Definition.MyBase) files\Git-2.8.4-32-bit.exe'

Download of 'Git-2.8.4-64-bit.exe' completed.
Replacing 'https://github.com/git-for-windows/git/releases/download/v2.8.4.windows-2.8.4-64-bit.exe' with '$(Split-Path -parent $MyInvocation.MyCommand.Definition.MyBase) files\Git-2.8.4-64-bit.exe'
Updating chocolateyInstall.ps1 with local resources.
Recompiling package.

Recompiled package files available at 'C:\packages\download\git.install'
Recompiled nupkg available in 'C:\packages'
```





How does it work?







Package Synchronizer (C4B)

- stay in sync w/system





Package Synchronizer (Licensed) - Auto Sync

```
PS C:\> choco list -lo
Chocolatey v0.9.10-beta1-422-g335bc01 Bu
[Synchronize] Removing 1password package
tey.
chocolatey 0.9.10-beta-20160609
chocolatey.extension 1.4.0
chocolatey.server 0.1.1
DotNet4.0 4.0.30319.20141222
git 2.7.0.0
git.install 2.8.4
GoogleChrome 51.0.2704.84
Launchy 2.5.0.20140301
roundhouse 0.8.5.0
9 packages installed.
PS C:\>
```

- Chocolatey maintains state based on packages. System state can be manipulated outside of Chocolatey
- Any Chocolatey command will trigger synchronization in licensed editions of Chocolatey
- Package Synchronizer syncs with manual software removal
- Syncs w/software that automatically upgrades, such as Chrome



Package Synchronizer (C4B) - Choco Sync Command

```
PS C:\> choco sync -r
Package Id|Display Name|Version|New Package
7zip|7-Zip 16.02 (x64)|16.2|False
oracle-vm-virtualbox-guest-additions|Oracle V
12.0|True
bitvise-ssh-server|Bitvise SSH Server 5.60 (r
PS C:\>
```

- Brings all software in Programs and Features under Chocolatey management
- Generates packaging and baselines to Chocolatey install
- Provides packaging code to take back to source control



Other Features

- <https://chocolatey.org/docs/release-notes>
- Internal sources (like ProGet)
- *choco upgrade all* - Windows update for your 3rd party and internal software
- Shimming - like symlinks but better
- Pass install arguments directly through to installer (append or override package args)
- Package Parameters to adjust logic in packages
- Handles locking on upgrades in package folders
- Great reference docs - <https://chocolatey.org/docs/commands-reference> & <https://chocolatey.org/docs/helpers-reference>
- Excellent open source support community
- 30+ built-in PowerShell Functions
- Extend functionality with your own PowerShell Modules
- Custom Package Templates
- Can manage anything on Windows





Other Features (C4B)

- <https://chocolatey.org/docs/release-notes-licensed>
- Auto-uninstaller more effective - uninstaller detection / Package Synchronizer
- Self-Service for non-administrators
- Install/upgrade keeping secrets out of logs
- Ubiquitous Install Directory Option
- More built-in PowerShell functions
- Uninstall non-Chocolatey managed software
- Direct Installer - install directly from exe/msi
- Package Reducer / Package Throttle
- Runtime malware protection
- CDN Cache
- Professional Packaging Services
- Amazing expert support team - <https://chocolatey.org/support>





Chocolatey Roadmap

- <https://chocolatey.org/docs/roadmap>
- Central Management UI (C4B)
- Choco Deploy CLI (C4B)
- WSA / Windows Nano Support
- PowerShell
PackageManagement Official Provider
- Package Enhancements
- Pack Validation
- Virtual packages
- GPG Signing
- Package Grid (C4B)
- Deferred Installer (C4B)
- Possible support on other platforms





Integration & Use Case





Organizational / Internal Use





Understanding Chocolatey and Packaging

Community Package Repository

- Public domain
- Distribution rights
- Has Internet dependencies - failure point
- Possible trust issues for organizations

Organizational Use

- Internal / Offline
- Many thousands of packages larger
- Reliability

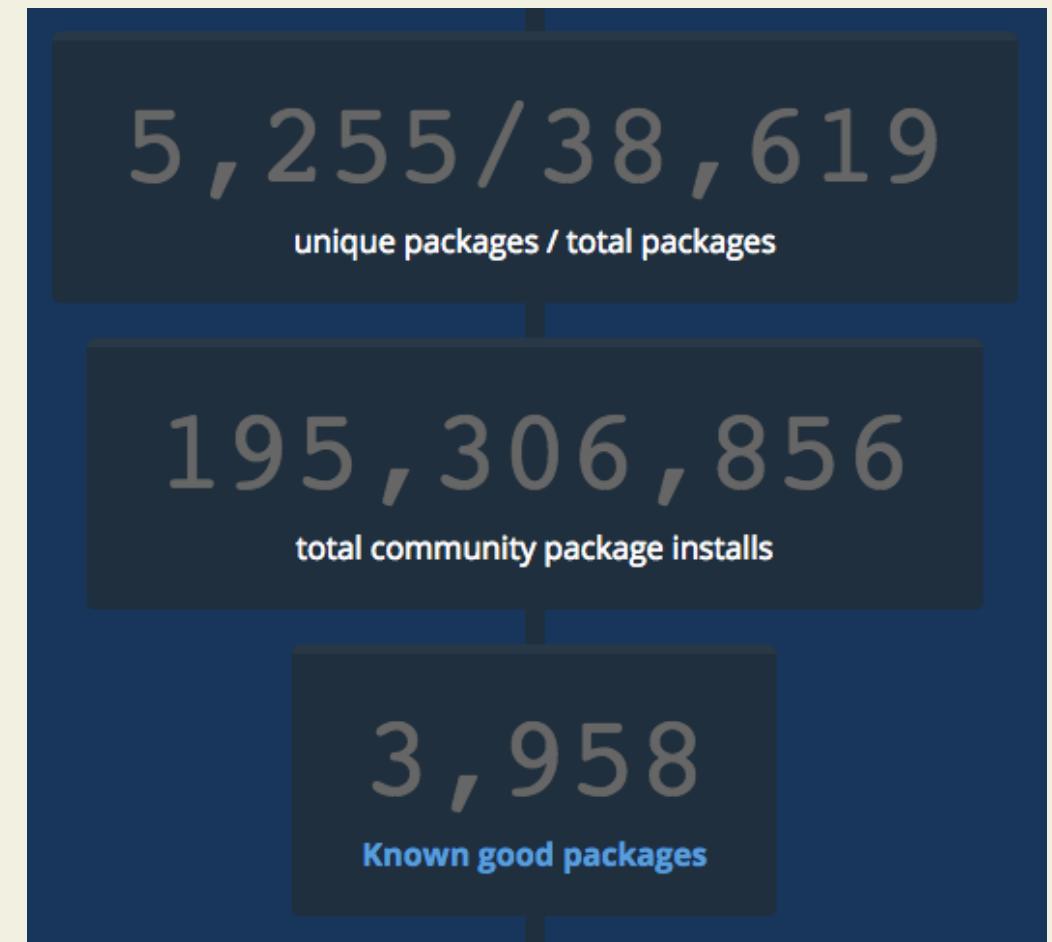
Do not cross this line to external / internet

WARNING: Objects in mirror may be larger than they appear



Community Package Repository

- <https://chocolatey.org/packages>
- Community Feed / Community Maintained
- Moderated as of October 2014
- Everything goes through VirusTotal
- **Organizations Should Avoid**
 - **Not Fully Reliable - public repo means subject to distribution rights (download CDN cache feature helps)**
 - **Trust and Control**





Artifactory Provides Reliability

- Create packages for internal use
- Internalize existing packages from community repository
 - Package Internalizer or manually
 - Recommend not using remote repository to community repo





Artifactory + Chocolatey

- Set up Local NuGet repository for now
- Ignore some of the NuGet specifics

General

When using Artifactory as a NuGet repository you can either work with the NuGet client directly or with Visual Studio

NuGet Client Configuration

To configure NuGet repository use the following command:

```
choco source add --name artifactory --source <url> --priority 1 --bypass-proxy --allow-self-service
```

```
1 nuget sources Add -Name Artifactory -Source http://localhost:8081/artifactory/api/nuget/chocolatey-local
```

To enable the use of NuGet API key, run the following command

```
1 nuget setapikey <USERNAME>:<PASSWORD> -Source Artifactory
```

```
choco apikey -s <full url> -k username:password
```

Deploy

Uploading packages to Artifactory can be done by running the following command:

```
1 nuget push <PACKAGE_NAME> -Source Artifactory
```

```
choco push <PACKAGE_NAME> --source <full url>
```

To support more manageable layouts and additional features such as cleanup, NuGet repositories support custom layouts. When pushing a package, you need to ensure that its layout matches the target repository's layout:

```
1 nuget push <PACKAGE> -Source http://localhost:8081/artifactory/api/nuget/+chocolatey-local/<PATH_TO_FOLDER>
```

Resolve

NuGet Client Resolve





Demo





Questions?

Start Your Free Trial

- C4B - <https://chocolatey.org/contact> (Sales)
- Artifactory - <https://www.jfrog.com/artifactory/free-trial/>



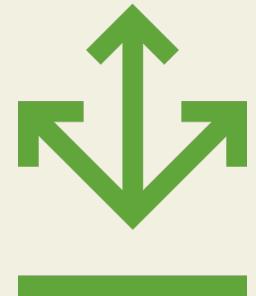
JFrog Artifactory





Join us again

- JFrog BinTray + Chocolatey
- Will be announced later



JFrog Bintray

