Simplify Windows Software Packaging and Automation with Chocolatey

Rob Reynolds
Founder, Chocolatey Software
@ferventcoder
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 14 years experience in infrastructure automation, mostly in Windows
• Obsesses over user experience
• Worked on configuration management for Windows for _another_ organization
Agenda

• Chocolatey
• Organizational Use and Ansible
• Demos
Be Honest, By A Show of Hands
How many of you...
...just came for the chocolate?
...are Linux Admins now supporting Windows?
...have no idea why you are in here?
...are here for Chocolatey?
Stand Up If You Use Chocolatey Currently

• Stay standing If You Started on...
  • v0.10.15
  • v0.10.0
  • v0.9.9
  • v0.9.8.33
  • v0.9.8.25
  • Older?
What is Chocolatey?
Chocolatey is...

- Windows software management / package management for Windows
- Universal management for zips, scripts, binaries, and installers
- Software deployment automation packages
With Chocolatey you can...

- Manage ANY software, not just installers
- Define dependencies
- Write a software deployment one time (with PowerShell!)
- Test your deployments - gain confidence long before a prod deployment
- Deploy to any supported version of Windows (including Server.Core / Docker containers)
- On prem, in the cloud, etc
- With any remote deployment tool (Ansible, DSC, SCCM, LanDesk, MDT, etc) or combination
- Track and Report on software
Chocolatey allows you to deploy any Windows software, anywhere, with anything, and manage and track that software over time.

Did we mention PowerShell?
Chocolatey is a Framework

- Lightweight
- Amazingly flexible
- Makes Windows software management very easy
- Catalyst for reducing automation work on Windows
Editions

- Open Source - DIY, Apache v2 license
- C4B (Chocolatey for Business) - More UIs, smoother experience, more precision, and features geared towards organizational use
- Easy to migrate between the two
Commercial Capabilities

Real-time Insights
Chocolatey Central Management

Accelerate Deployments
Package Builder & Package Uploader

Improve Security & Controls
Package Internalizer & Package Audit

Improve Accuracy & Simplify Software Management
Package Synchronizer

Enhance User Experience
Self-Service GUI

Optimization & Environments
Package Reducer & Package Throttle
So what is Chocolatey?
It is software management that works
Why Chocolatey? Because the following is NOT how we should manage software
Organizational Use
Organizational Use

Packaging
choco new, Package Builder, internalization

Source Control
Git & Git LFS

Internal Repositories
Artifactory, Nexus, ProGet

Continuous Integration
Jenkins, Team City, etc

Deployment & Management
Ansible, SCCM, etc

Reporting
Current, Central Management, DIY
Repositories
**Chocolatey Architecture Diagram**

* Clients could be managed directly or through end point management solutions like Microsoft SCCM, Microsoft Intune, Altiris, LanDesk, etc. Also Chocolatey's Central Management.
Why Can’t I Just Use The Community Repository?
Community Package Repository

- [https://chocolatey.org/packages](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

<table>
<thead>
<tr>
<th>Unique Packages / Total Packages</th>
<th>7,169/76,145</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Community Package Installs</td>
<td>672,255,271</td>
</tr>
<tr>
<td>Known Good Packages</td>
<td>5,300</td>
</tr>
</tbody>
</table>
Internal Repositories

- [https://chocolatey.org/docs/how-to-host-feed](https://chocolatey.org/docs/how-to-host-feed)
- Artifactory Pro (cross-platform)
- Sonatype Nexus (cross-platform)
- Inedo’s ProGet
- MyGet (hosted)
- Chocolatey.Server
- NuGet.Server / NuGet Gallery proper
- Visual Studio Team Services (v2 endpoints) / TFS
- File Share (DFS, CIFS, SMB) / SCCM Distribution points

Commercial options:
- Web Interface
- One Install / Multiple repositories
- Multiple repository types, not just NuGet/Chocolatey
Packaging
Package Internalizer - “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
Creating Packages

- [https://chocolatey.org/docs/create-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- “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 for 70% of installers out there
- Supports passing urls (and checksums) - downloads files first
- Allows customization
- Supports exe, msi, 7z, zip, msu, and msp files
Non-Silent Installers?

• MSI Repackaging
• C4B Customers can use optional Packaging Services for small fee
Deployment & Management
* Clients could be managed directly or through endpoint management solutions like Microsoft SCCM, Microsoft Intune, Altiris, LanDesk, etc. Also Chocolatey's Central Management.
Infrastructure Management

- [https://chocolatey.org/docs/features-infrastructure-automation](https://chocolatey.org/docs/features-infrastructure-automation)
- **Tight - provider-based**
  - Ansible
  - Others
- **Script-based**
  - Octopus Deploy
  - Boxstarter
  - SCCM
  - RMM, or really anything that can run scripts
Provider Integrations

- In the provider’s language:
  - Provide Chocolatey installation
  - Provide configuration
  - Provide software management

```yaml
- name: Ensure Chocolatey installed from internal repo
  win_chocolatey:
    name: chocolatey
    source: http://some server/chocolatey/Chocolat eyInstall.ps1

- name: Set the choco cache location
  win_chocolatey_config:
    name: cacheLocation
    state: present
    value: C:\ProgramData\choco-cache

- name: Increase choco timeout to 4 hours
  win_chocolatey_config:
    name: commandExecutionTimeOutSeconds
    state: present
    value: 14400

- name: Disable download progress
  win_chocolatey_feature:
    name: showDownloadProgress
    state: disabled

- name: Remove the default choco public source
  win_chocolatey_source:
    name: chocolatey
    state: absent

- name: Set new default internal choco source
  win_chocolatey_source:
    name: internal_repo
    state: present
    source: https://chocolatey-server/chocolatey
    source_username: optional
    source_password: optional
    priority: 1
```
Ansible - Installing Chocolatey: win_chocolatey

```yaml
- name: Ensure Chocolatey installed from internal repo
  win_chocolatey:
    name: chocolatey
    source: http://someserver/chocolatey/ChocolateyInstall.ps1
```

• Look at [https://chocolatey.org/install](https://chocolatey.org/install) and select Organization

1. Download the Chocolatey package

2. Put the Chocolatey package on your internal repository

   Please see hosting packages internally and the organizational deployment guide.

3. Enter your internal repository url

   http://Internal/odata/repo

   (this should look similar to [https://chocolatey.org/apiv2](https://chocolatey.org/apiv2))
Prepping For Internal Deployment

• Internal Chocolatey install script
• Organizational Deployment Guide

1. Download the Chocolatey package

2. Put the Chocolatey package on your internal repository
   Please see hosting packages internally and the organizational deployment guide.

3. Enter your internal repository url
   [http://internal/odata/repo](http://internal/odata/repo)
   (this should look similar to [https://chocolatey.org/api/v2](https://chocolatey.org/api/v2))

4. Use this install script instead

5. Make script accessible by internal url
   Put the script somewhere internally that can be accessed by a url (like a raw/binary repository)

6. Install/Deploy Chocolatey internally
   Now you can do similar to the individual side, but your command will be something like:

   ```powershell
   ```

7. Ensure you are set up for organizational deployment
   Please see the organizational deployment guide.
Ansible - Package Management: win_chocolatey

• Provides software management
• Lots of options
• Become

- name: Ensure git installed
  win_chocolatey:
    name: git
    state: present
    version: 2.23.0
    source: http://internal/odata/repo
    source_username: optional
    source_password: optional

- name: Install a package that requires 'become'
  win_chocolatey:
    name: officepro2013
    become: yes
    become_user: Administrator
    become_method: runas

- name: install and pin Notepad++ at 7.6.3
  win_chocolatey:
    name: notepadplusplus
    version: 7.6.3
    pinned: yes
    state: present
Ansible - Sources: win_chocolatey_source

- Remove default community source

```
- name: Remove the default choco.public.source
  win_chocolatey_source:
    name: chocolatey
    state: absent
```

- Set your internal sources

```
- name: Set new default internal.choco.source
  win_chocolatey_source:
    name: internal_repo
    state: present
    source: https://internal/odata/repo
    source_username: optional
    source_password: optional
    priority: 1
```
Ansible - Features: win_chocolatey_feature

- Turn on and off features
- [https://chocolatey.org/docs/chocolatey-configuration](https://chocolatey.org/docs/chocolatey-configuration)

```yaml
- name: Disable download progress
  win_chocolatey_feature:
    name: showDownloadProgress
    state: disabled

- name: Exit when reboot is needed
  win_chocolatey_feature:
    name: exitOnRebootDetected
    state: enabled
```
Ansible - Config: win_chocolatey_config

• Set configuration settings

• [https://chocolatey.org/docs/chocolatey-configuration](https://chocolatey.org/docs/chocolatey-configuration)
Reporting
Exhibit 1: Sample output from the `choco list -1o --include-programs` command with `--audit` flag.

- **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 adds who, when, and history**
Compliance – Audit Out of Date Software

- choco outdated
- Comprehensive reporting on available upgrades
- See what will upgrade on next maintenance window
- Pinned items won't upgrade until pin is remove
Package Synchronizer - Auto Sync

- 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
Chocolatey Central Management

- See reports / information across the org
- Modules / plugins coming
- Deployments
Other Features

- [https://chocolatey.org/docs/release-notes](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
- 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-notesLicensed](https://chocolatey.org/docs/release-notes-Licensed)
- Auto-uninstaller more effective - uninstaller detection / Package Synchronizer
- 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 Throttle
- Runtime malware protection
- CDN Cache
- Professional Packaging Services
- Amazing expert support team - [https://chocolatey.org/support](https://chocolatey.org/support)
Chocolatey Roadmap

- [https://chocolatey.org/docs/roadmap](https://chocolatey.org/docs/roadmap)
- Choco Deploy CLI (C4B)
- WSA / AppX / MSIX
- Package Enhancements
- Pack Validation
- More GUIs (C4B)
- Authenticode / GPG Signing
- Package Grid (C4B)
- Deferred Installer (C4B)
- Others (just ask)
Learn More

- [https://chocolatey.org/contact](https://chocolatey.org/contact) (choose “Sales”)
- [https://chocolatey.org/docs](https://chocolatey.org/docs)
- Expansive community
- chocolateynuget on twitter
Organizational Use

- **Packaging**
  - choco new, Package Builder, internalization

- **Source Control**
  - Git & Git LFS

- **Internal Repositories**
  - Artifactory, Nexus, ProGet

- **Continuous Integration**
  - Jenkins, Team City, etc

- **Deployment & Management**
  - Ansible, SCCM, etc

- **Reporting**
  - Current, Central Management, DIY
Anyone here doing all of this?
INTERNET
HIGH-FIVE
PLACE HAND
HERE
Demos
Other Windows Talks Here

Today
• 3PM Rm 204-205: 1000 Windows Servers on a 30 day rebuild!
  • Speaker(s): Nathan Michael, G-Research; Nic McElroy, G-Research
• 4PM in 304-305: 10 Things I Hate About You - Manage Windows Like Linux with Ansible
  • Speaker(s): Matt Davis, Red Hat

Tomorrow
• 10AM Rm 204-205: Ansible Tower and Windows Management
  • Speaker(s): James Bailey, Worldpay Powered FIS; Sam Pusey, Worldpay
• 2PM Rm 304-305: Me, Myself, and System - Using Become on Windows
  • Speaker(s): Jordan Borean, Ansible

Other talks may include Windows
Stop By Our Booth!

- Random drawings for Config Mgmt Strawberry T-Shirts - based on folks who have stopped by and have been scanned
- More Chocolatey Grey T-Shirts
- Meet our folks
- See a Demo
- Tell us about your Windows experience!
#ANSIBLEFEST2019

THANK YOU

youtube.com/AnsibleAutomation
facebook.com/ansibleautomation
linkedin.com/company/Red-Hat
twitter.com/ansible