

# ActiveState

## Retaining Your Software Developers

Keeping Software Devs Happy by  
Automating Open Source Pains



# Pandemic-Created Pressures

### **Increased developer workload/stress as a result of:**

- The need to enable a remote workforce “overnight”
  - Accelerated digital migration plans

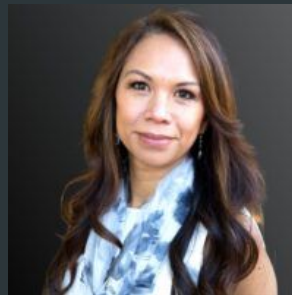
### **“Work from anywhere” trend:**

- Developers have more employment opportunities than ever before
  - The need to onboard new developers quicker

# Introductions



Shane Warden  
Director of Engineering



Loreli Cadapan  
VP Product Management

# The Open Source Opportunity


- Time consuming tasks
  - ie, Building dependencies from source
- Frustrations
  - ie, Dependency Hell
- Avoidable Risks
  - ie, Supply chain security


Eliminating these can reduce  
developer workload and stress

Help Developers Save Time


## Automate Builds from Source


- Automatically build open source from source code + native libraries

 **Windows 10** ⓘ  
x86 · 64-bit

 **Linux Glibc 2.28** ⓘ  
x86 · 64-bit

**Build Status** ... IN PROGRESS Elapsed time: 0m 55s **26 of 46 Packages Built**

 Your build may take 20 minutes or more. We'll notify [danac@activestate.com](mailto:danac@activestate.com) when it's ready (or if there's an error) - [Learn more about our build process and why we build from source.](#) ⓘ



▼ Package build status


**Building** 15

chardet	3.0.4	<b>Building</b>	Build time - 0m 43s	> View Logs
cppy	1.1.0	<b>Building</b>	Build time - 0m 43s	> View Logs
cycler	0.10.0	<b>Building</b>	Build time - 0m 43s	> View Logs

## Automate Vulnerability Management

- Automatically monitor, notify and simplify remediation of vulnerabilities

Ruby 1 Package Vulnerabilities (CVEs)

rails  Auto (7.0.2) 2 CVEs

● Medium





[CVE-2022-23634](#)

[CVE-2022-23633](#)

**Dependencies 57**  
Automatically added to support reques


Custom >

Auto

	7.0.2	2 CVEs
	7.0.1	2 CVEs
	7.0.0	3 CVEs
	6.1.5	0 CVEs

# Eliminate Locating Dependencies

- Automatically create a registry of known, good dependencies that work together

Python 3 Packages		Vulnerabilities (CVEs)		Licenses
django	 3.2.13 	 0 CVEs		
requests	 2.25.0 	 0 CVEs		
torch	 0.1.2 	 0 CVEs		



# Reduce Developer Frustration

## Eliminate Dependency Hell



There's a problem with some of the packages!

We were unable to provide some of the packages or their dependencies. Until the issue is resolved, we won't be able to start a build for this project. View the details below for more information.

▼ View details

Because `Feature|language/ruby|rails (7.0.2)` requires `Ingredient|language/ruby|rails (7.0.2)` which depends on `Feature|language/ruby|actionmailer (7.0.2)`, `Feature|language/ruby|rails (7.0.2)` requires `Feature|language/ruby|actionmailer (7.0.2)`.

And because `Feature|language/ruby|actionmailer (7.0.2)` requires `Ingredient|language/ruby|actionmailer (7.0.2)` which depends on `Feature|language/ruby|rails-dom-testing (>=2.0,<3)`, `Feature|language/ruby|rails (7.0.2)` requires `Feature|language/ruby|rails-dom-testing (>=2.0,<3)`.

So, because `root` depends on both `Feature|language/ruby|rails (7.0.2)` and `Feature|language/ruby|rails-dom-testing (1.0.9)`, version solving failed.

## Automate Environment Reproducibility

- Eliminate “works on my machine” issues & ensure environment reproducibility

The screenshot displays the ActiveState web interface. On the left, under the 'Details' tab, the environment configuration is listed: Python 3.9.12 and 8 Packages. A 'Configure' button is visible. On the right, the 'Builds' section shows three operating systems: Linux Glibc 2.28, Mac, and Windows 10. Each entry includes an icon, the OS name, a status of 'Build ready' in green, and an 'Install' button with a download icon.

OS	Status	Action
Linux Glibc 2.28	Build ready	Install
Mac	Build ready	Install
Windows 10	Build ready	Install

## Manage Runtime Configuration Drift

- Parent-child branches inherit changes between dev, test, prod, etc environments

The screenshot shows the 'pyproject' configuration page in the ActiveState interface. The top navigation bar includes 'Overview', 'Configuration' (highlighted), 'Download Builds', 'History', 'Project Settings', and 'Admin'. The 'Configuration' section shows a 'Branch' dropdown set to 'main / production'. A 'Manage' button is next to it. A notification bubble states: 'Changes available from upstream branch — Update'. Below this, a dropdown menu shows 'main' and 'production' (selected). The 'Language' section shows 'Python 3.9.12' with a 'Change Version' button. The 'Vulnerabilities' section shows '4 Critical' and '1 High'. The 'Requested Packages' section is partially visible.

### Update your branch

You are about to update your branch with the latest changes made to the upstream branch.

#### Changes

















Adding + pytorch Auto

# Help Developers Avoid Risk

# Vetted Dependency Catalog

- Understand security and licensing risks at a glance

Python 9 Packages		Vulnerabilities (CVEs)		Licenses
keras	 2.0.8	 0 CVEs		
matplotlib	 1.2.0	 0 CVEs		
nltk	 3.4	 3 CVEs >		
pandas	 0.25.3	 1 CVE >		

		Vulnerabilities (CVEs)		Licenses
keras	 2.0.8	 5 licenses		
matplotlib	 1.2.0	 BSD-3-Clause		
nltk	 3.4	 CC-BY-4.0		
pandas	 0.25.3	 MIT		
		 OFL-1.1		
		 Python-2.0		

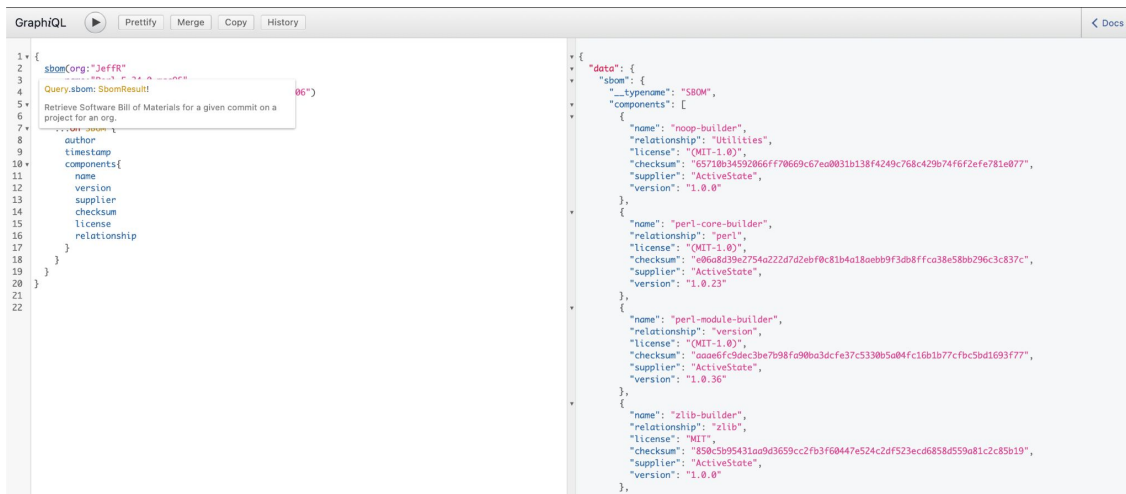
## Secure Build Service

- Tamper-proof system creates reproducible builds of secure artifacts

Build	Scripted	✓
	Build Service	✓
	Ephemeral Environment	✓
	Isolated	✓
	Parameterless	✓
	Hermetic	✓
	Reproducible	✓

# Automatically Generate SBOMs

- Software Bill Of Materials (SBOM) for each of your runtime environments



The screenshot shows the GraphQL Playground interface. On the left, a query is entered: `query sbom { sbom(org: "JeffR") { components { name version supplier checksum license relationship } } }`. A tooltip for the `sbom(org: "JeffR")` field shows the description: "Retrieve Software Bill of Materials for a given commit on a project for an org." On the right, the JSON response is displayed, showing a list of components with their names, versions, suppliers, checksums, licenses, and relationships.

```
1 {
2   sbom(org: "JeffR")
3 }
4
5 Query: sbomResult!
6
7 Retrieve Software Bill of Materials for a given commit on a
8 project for an org.
9
10 author
11 timestamp
12 components {
13   name
14   version
15   supplier
16   checksum
17   license
18   relationship
19 }
20 }
21
22
```

```
{
  "data": {
    "sbom": {
      "__typename": "SBOM",
      "components": [
        {
          "name": "noop-builder",
          "relationship": "Utilities",
          "license": "MIT-1.0",
          "checksum": "65718b34592866ff70669c67ea0831b138f4249c768c429b74f672e781e077",
          "supplier": "ActiveState",
          "version": "1.0.0"
        },
        {
          "name": "perl-core-builder",
          "relationship": "perl",
          "license": "MIT-1.0",
          "checksum": "e96a8d39e2754a222d7d2ebf0c81b4a18aebb9f3db8ffca38e58bb296c3c837c",
          "supplier": "ActiveState",
          "version": "1.0.23"
        },
        {
          "name": "perl-module-builder",
          "relationship": "version",
          "license": "MIT",
          "checksum": "a9a6fcd3e3be798fa90ba3dcfe37c5338b5a04fc16b1b77cfbc5bd1693f77",
          "supplier": "ActiveState",
          "version": "1.0.36"
        },
        {
          "name": "zlib-builder",
          "relationship": "zlib",
          "license": "MIT",
          "checksum": "890c595431a9d3659cc2fb3f60447e524c2df523ecd6858d559a81c2c85b19",
          "supplier": "ActiveState",
          "version": "1.0.0"
        }
      ]
    }
  }
}
```



Poll: To what extent do you automate your use of open source dependencies?

- Not at all
- Minimal automation (ie., automated import to repo)
- Some automation (ie., import + build system)
- Mostly automated (ie., import, build & vulnerability remediation)
- End to end automation (ie., import, build, vulnerabilities & environment setup/update, SBOM generation, etc)

# Pros and Cons of Self-Vendoring

### Pros:

- Avoid Dependency Conflicts
- Avoid Breaking the Build
- Consistent Deployments
- Faster Fixes

### Cons:

- Need to Build Everything Yourself
- Outdated Dependencies
- Invisible Vulnerabilities
- Source Code/Repository Clutter

## ActiveState's Managed Distributions

- Produce a catalog of dependencies
- Securely built
- Reproducible runtime environments across all operating systems
- Optionally distribute via artifact repositories
- Monitor dependencies for vulnerabilities and stale-datedness
- Optionally apply security fixes for EOL languages (e.g. Python 2)

# Platform Demo

# Demo: How to Onboard New Developers Quicker

- A cross-platform project:  
<https://platform.activestate.com/shanew/keeping-devs-happy-demo-june-2022>
- Built on Linux, but let's activate and run it elsewhere

## Demo: Import Packages

Import Packages

Close

Paste the contents of your requirements.txt, Pipfile, Pipfile.lock, pyproject.toml or poetry.lock file below.

File Contents

flask  
numpy  
pillow

Import

## Demo: Dependency Hell Solved Automatically

You have unsaved changes. Save your changes to update your project.

Importing packages from Python project file
43/100
Save
Cancel

Language
Change Version

Python 3.9.12

1 CVE
>

Platforms
Change

Linux Glibc 2.28 ⓘ

x86 · 64-bit

Mac ⓘ

x86 · 64-bit

Windows 10 ⓘ

x86 · 64-bit

Vulnerabilities

2 Critical
1 High

Requested Packages
Add Packages
Import from File

Python 3 Packages

Vulnerabilities (CVEs)
Licenses

+ flask	Auto (2.1.2) ▾	0 CVEs	Cancel
+ numpy	Auto (1.22.1) ▾	0 CVEs	Cancel
+ pillow	Auto (9.1.0) ▾	0 CVEs	Cancel

Dependencies 38

Automatically added to support requested packages & platforms.

26 Changes
☐ Only show changes

## Demo: Built on All Platforms

The screenshot shows the ActiveState interface for a project named 'keeping-devs-happy-demo-june-2022' by user 'shaney'. The project is a 'Public Project'. The navigation bar includes 'Overview', 'Configuration', 'Download Builds' (highlighted), 'History', 'Project Settings', and 'Superuser™ Fork'. On the left, three platform build cards are visible: 'Windows 10 x86 - 64-bit', 'Mac x86 - 64-bit', and 'Linux Glibc 2.28 x86 - 64-bit'. The main content area shows the 'Build Status' as 'Finished' with a green progress bar and '39 Packages Built'. Below this, there is a section for 'First time install via command line' with a terminal command to install the State Tool package manager. A note explains that this command installs the State Tool package manager. Another section, 'Installed it already? Get the latest changes', provides a command to update the local installation.

shaney  
keeping-devs-happy-demo-june-2022 Fork It Public Project

Overview Configuration Download Builds History Project Settings Superuser™ Fork

Windows 10 x86 - 64-bit

Mac x86 - 64-bit

Linux Glibc 2.28 x86 - 64-bit

Build Status **Finished** Build time - 2m 31s 39 Packages Built

> Package build status

First time install via command line

```
sh <(curl -q https://platform.activestate.com/dl/cli  
/1194273648.1628270787_pdl1a01/install.sh) -c'state activate --default  
shaney/keeping-devs-happy-demo-june-2022'
```

This command installs the State Tool package manager, which lets you manage your runtime. Find tips on managing your runtime in the CLI with our [State Tool Cheat Sheet](#)


Installed it already? Get the latest changes

If you've already installed this runtime, run the following command to update your local installation to use the latest build.

```
state pull
```



## Demo: Associated With Your Repository

 shanew

keeping-devs-happy-demo-june-2022

Fork It

Private Project

Overview

Configuration

Download Builds

History

Project Settings

Superuser™ Fork

Project Details

Branches

Details

Name

keeping-devs-happy-demo-june-2022

Description (Optional)

Git Repo URL (Optional)

When activating this runtime via the command line for the first time, this repo will automatically be cloned into the virtual environment.

https://github.com/swagile1/flask-demo

**ActiveState**

Q&A

## Next Steps

Schedule a demo with our product experts:

<https://www.activestate.com/get-demo/>

Learn more about Managed Distributions:

<https://www.activestate.com/blog/outsourced-dependency-vendoring/>

Try the ActiveState Platform for free:

<https://platform.activestate.com/>

# Webinar Feedback

Take our quick survey!

<https://www.surveymonkey.com/r/keeping-devs-happy>