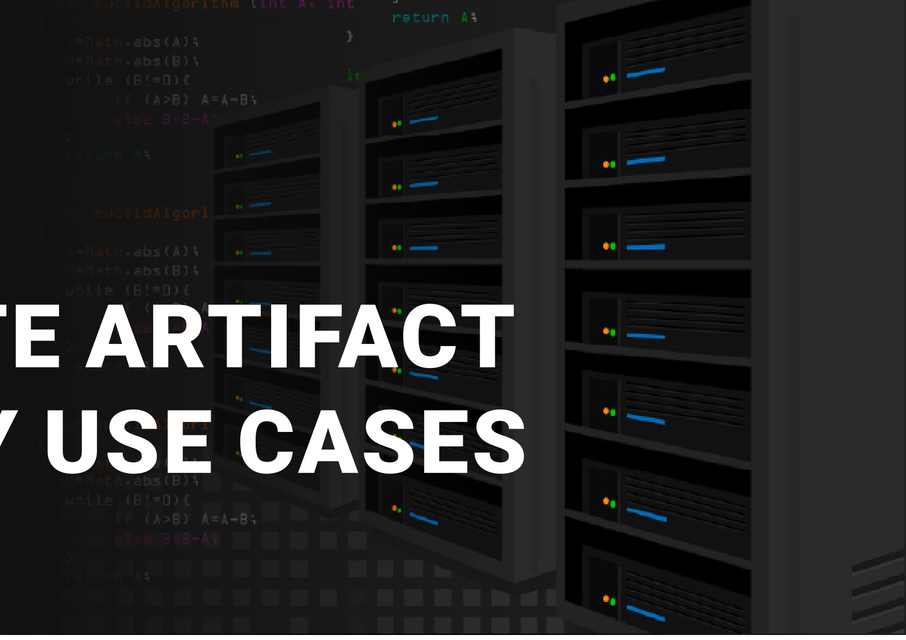


ACTIVESTATE ARTIFACT REPOSITORY USE CASES

Artifact repositories store built software artifacts and make them available to users, systems and processes. They serve a variety of purposes, including:

- **Development:** provides a central location to manage a curated set of approved open source and other third-party artifacts used in the development process.
- **CI/CD:** provides a place to store built artifacts output from the CI/CD system, as well as a source to pull inputs from, such as prebuilt containers or runtimes.
- **Production:** provides a central location for distributing built software artifacts to customers. This includes patches, updates and applications.

The ActiveState Artifact Repository focuses on third-party, open source code, allowing organizations to better manage and distribute them in a secure, consistent, repeatable manner. Some of the key use cases for the ActiveState Artifact Repository include:

1. **Fostering Open Source Supply Chain Security**
2. **Creating a Centralized, Curated Catalog**
3. **Standardizing Application Extensibility**
4. **Simplifying Python Wheel Maintenance**

The ActiveState Artifact Repository currently provides support for:

- Secure, multi-platform, cloud-based builds that eliminate the need to set up and maintain local Windows, Mac and Linux build systems.
- Python 3.8+ artifacts built automatically from source code, including linked C libraries.

#1. Open Source Supply Chain Security

The Problem: Unsigned Binary Artifacts in Public Repositories

Open source public repositories like the Python Package Index (PyPI) are popular because their "openness" allows anyone to publish any code they want, which software vendors can then reuse with trivial effort thanks to the ecosystem's package manager. But this very openness is what makes the open source ecosystem so vulnerable to supply chain exploits, because open source repositories do not have:

- Strict controls around **typosquatting and brandjacking**, which can result in developers mistakenly downloading compromised packages.
- Strict guidelines to ensure that prebuilt packages are securely built.
- Signed artifacts, which would ensure that prebuilt packages have not been tampered with after they have been built.

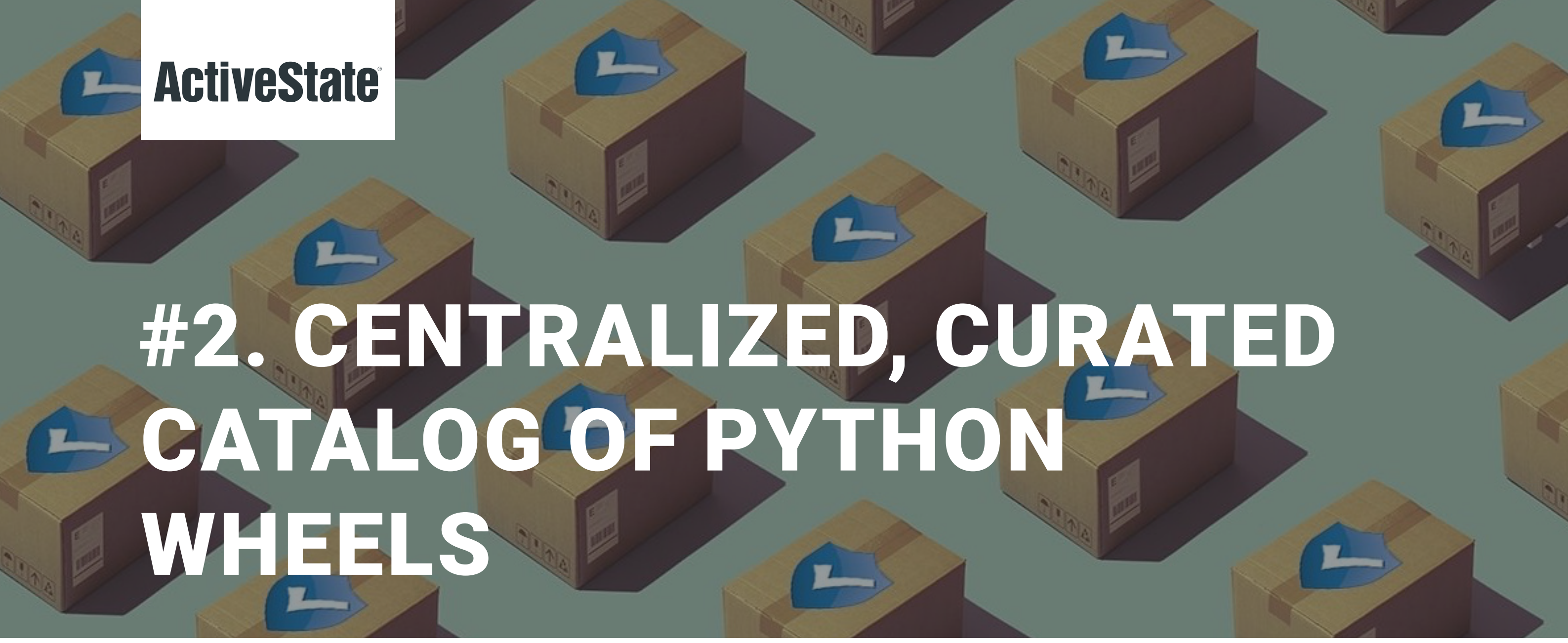
The Solution: Trusted Binary Build Service

The ActiveState Artifact Repository is the industry's only artifact repository that includes a secure build service, which automatically builds Python wheels with all the security and integrity controls defined in the Supply chain Levels for Secure Artifacts (SLSA) specification, including:

- A local catalog of Python source code vetted on import from public repositories.
- Build scripts that cannot be accessed and modified by users, preventing exploits.
- Ephemeral, isolated build steps that execute in their own containers, which are discarded at the completion of each step.
- Hermetically sealed environments that have no internet access, preventing (for example) dynamic packages from including remote resources.
- Reproducible builds where the build process fails safe, terminating the build if any component generated during a build step fails its checksum verification.

The Benefits: Ensure Security & Integrity of Open Source Components

- Eliminate the risk of working with binary artifacts imported from public repositories.
- Work only with open source components securely built from vetted source code.
- Eliminate the need to periodically audit internal build systems for compromise by utilizing a secure, cloud-based build service.



ActiveState

#2. CENTRALIZED, CURATED CATALOG OF PYTHON WHEELS

The Problem: Developers Using Unapproved Dependencies

Most enterprises require that a lengthy process be navigated before a new dependency can be approved for use by developers. For example, each new dependency will need to be assessed by security personnel for vulnerabilities, development managers for maintainability, and compliance personnel for proper licensing. Only once it has passed all of these rigorous, time-consuming reviews can the dependency be securely built and finally made available.

However, developers are creative problem solvers. If they require a third-party library to ensure their deliverables are completed on time, they will find a way to route around the approval process and download the dependency directly, potentially introducing licensing, security and compliance issues.

The Solution: Walled Garden of Dependencies

Populating the ActiveState Artifact Repository starts with source code imported from the Python Package Index (PyPI), GitHub, and other locations. It is then vetted for:

- **Security** - source code is scanned for malware and checked for vulnerabilities.
- **Licensing** - licenses are displayed for each dependency, including the package-level license and sub-licenses, where available.
- **Maintainability** - historical releases are displayed, along with the number of unresolved vulnerabilities for each.

The ActiveState Artifact Repository acts as a single, central location for all stakeholders to evaluate new dependencies quickly, and then automatically build and make them available for use, dramatically shortening the approval and availability processes.

The Benefits: Ensure the use of Approved Dependencies

- Decrease time to approve and make new dependencies available for use.
- Decrease compliance, security and licensing risks.
- Increase Python dependency visibility for all stakeholders, from product and development managers to licensing and compliance auditors.

ActiveState

#3. STANDARDIZE APPLICATION EXTENSIBILITY

The Problem: Customer Support Costs

Software vendors that ship software to customers often provide the capability to extend their application using custom code. This enables customers to better integrate it with existing systems, or customize the solution for a specific use case.

But this also means the software vendor will be the first line of support for customers when their custom code fails. Because "custom code" can literally be anything, determining the point of failure is a non-trivial exercise that can dramatically inflate support costs.

The Solution: Standardized Extensibility

The ActiveState Artifact Repository can be made publicly available to all customers, acting as a cloud-based distribution platform for all Python dependencies approved for use. Rather than installing any package from PyPI, customers can be assured of technical support by working only with the set of approved third-party libraries sourced from the ActiveState Artifact Repository.

Software vendors that ship ActiveState's Python within their application installer can be assured that:

- Every customer installs a supported version of Python properly, in the same location.
- All Python packages installed from the ActiveState Artifact Repository will work with the pre-installed version of Python, and will work together without conflicts.

The Benefits: Empower Customers Without Increasing Support Costs

- Dramatically decrease support costs by providing a standard Python installation.
- ABI-compatible, prebuilt binary libraries are guaranteed to work on every customer's Windows, Mac, or Linux system.
- Easily reproduce issues without needing to troubleshoot the environment configuration.

#4. SIMPLIFIED PYTHON WHEEL MAINTENANCE

The Problem: Unudpated Codebases Increase Risk

Numerous reports and surveys repeatedly confirm the fact that once a codebase has been created, it's almost never updated. This shouldn't be a surprise since upgrades all too often result in breaking the build, typically due to the fact that upgrading one dependency triggers updates to other dependencies and/or transitive dependencies. Common build failures include:

- Dependency conflicts.
- Missing requirements.
- Build environment(s) that require updating to support the latest build configuration.

As a result, developers waste time and effort just to get back to the status quo, rather than focusing on their deliverables. On the other hand, unupdated codebases increase security and performance/ stability risks as unresolved vulnerabilities and bugs accumulate.

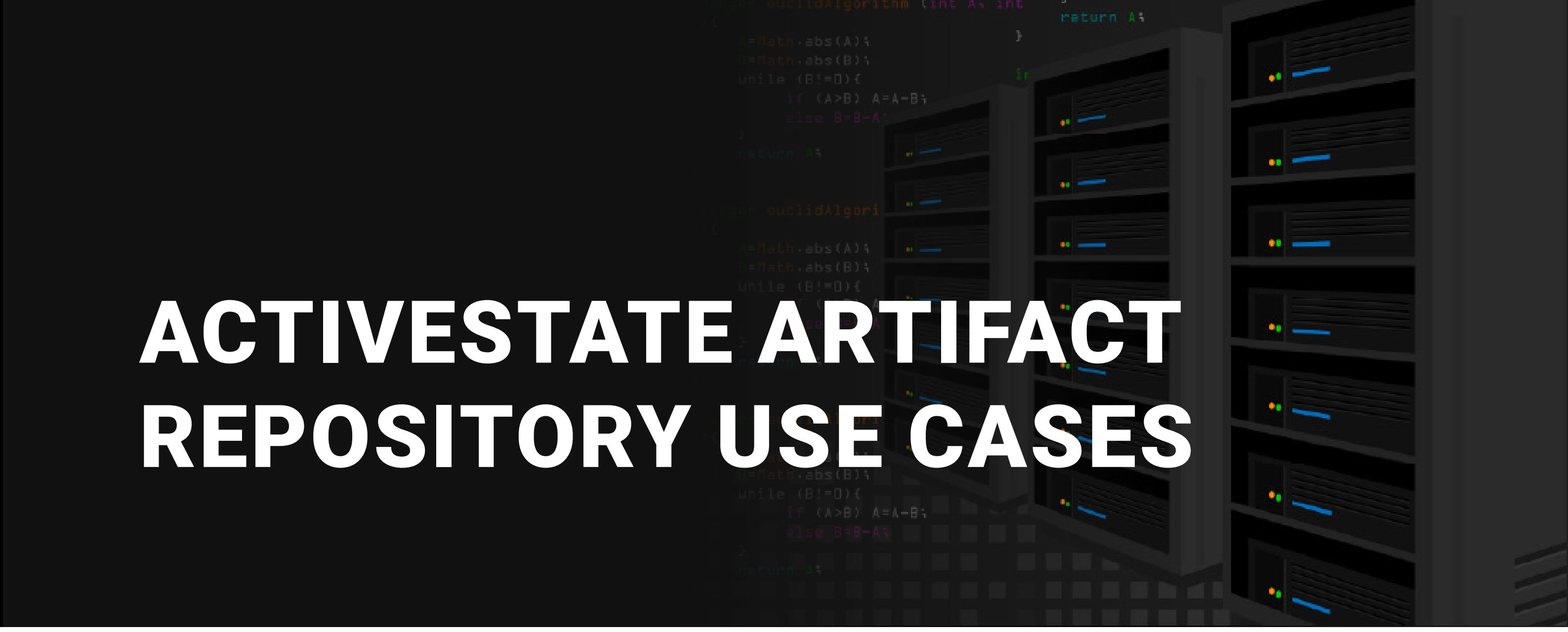
The Solution: Managed Distributions

ActiveState can maintain your Python wheels on your behalf. Once you've established a set of dependencies in the ActiveState Artifact Repository, ActiveState will:

- Monitor them for vulnerabilities and stale-datedness, and create a new repository with the updated dependencies for you to test at any time.
- Vet your dependencies from our catalog of third-party dependencies to ensure security, maintainability and appropriate licensing according to your corporate guidelines.
- Securely build your set of required Python wheels from source code, including native libraries for Windows, Mac and Linux.
- Maintain a catalog of all wheels and their dependencies over time so you can always reproduce the build.

The Benefits: Secured Codebase

- Reduce the costs associated with vendoring dependencies.
- Decrease risks associated with unupdated codebases.
- Recover lost time and resources previously spent managing and maintaining vendored dependencies.



ACTIVESTATE ARTIFACT REPOSITORY USE CASES

About ActiveState

ActiveState is the de-facto standard for millions of developers around the world who have been using our commerciallybacked, secure open source language distributions for over 20 years. With the ActiveState Platform, developers can now automatically build their own open source artifacts and environments—all without requiring language or operating system expertise.

Learn more about how the ActiveState Artifact Repository can help ensure your developers always work with secure Python dependencies.

Learn more

Book a demo and let us show you how the ActiveState Artifact Repository is the missing link in your Python Supply Chain:

Contact Sales

ActiveState

www.activestate.com
Toll-free in NA: 1-866.631.4581
solutions@activestate.com