(https://plus.google.com/117596333621717490325)

Risk Assessments (/resources/risk-assessments) Blog (http://blog.sonatype.com/people/)

Community (http://www.sonatype.org) Support (https://support.sonatype.com/home)

Search (/cart) (http://blog.sonatype.com/people/)

(http://www.sonatype.com/newsletter/latest)

(http://www.linkedin.com/company/sonatype) (http://twitter.com/Sonatype)

(http://www.facebook.com/sonatype) (http://www.youtube.com/user/sonatype)

Sonatype (/)

Nexus Pro Repository Manager

Maven and Nexus Pro, Made for Each Other

Maven has become instrumental in building modern day software by combining components and packaging them together. Applications or components that are created with Maven can be easily shared with other development teams.

Maven leverages the concept of a repository by retrieving the artifacts necessary to build an application and deploying the result of the build process into a repository. Maven uses the concept of structured repositories so components can be retrieved to support the build. These components or dependencies include libraries, frameworks, containers, etc. Maven can identify components in repositories, understand their dependencies, retrieve all that are needed for a successful build, and deploy its output back to repositories when the build is complete.

Developers using Maven without a repository manager find most of their software artifacts and dependencies in the Central Repository. If they use another remote repository or if they need to add a custom artifact, the developer has to manually manipulate the files in a local repository and share this local repository with multiple developers. While this approach may yield a working build for a small team, managing a shared local repository doesn't allow an organization to scale a development effort. There is no inherent control over who can set up a local repository, who can add to them or change or delete from them, nor are there tools to protect the integrity of these repositories.

Combining Maven – and other build tools -- with Nexus Pro, allows you to scale your development effort and build and share your component-based applications more effectively. With Nexus Pro, common frustrations are eliminated and new features speed development times. For example:

- Nexus supports a variety of build tools, repository formats and runtime environments

- Dramatically simplifies / saves time and effort
- Facilitates collaboration between different teams
- Reduces complexity / allows you to benefit from use of components
- Support heterogeneous development efforts
- Manage access to components based on roles
- Share components effectively with internal and external developers
- Provides natural starting point for full component lifecycle management

How does using Nexus Pro provide these benefits?

Easily share components with other developers - Nexus becomes the deployment destination for the components that are created by your organization. This way you can publish your own library or project JARs, WARs, EARs etc. using Nexus Pro. You can host your own repository to support your custom components and share those components with other users or computers on your network. Once your components are placed in the repository - they can be made available to all developers securely using access control and SSL delivery, so they don't have to download the components over and over. This makes it a lot easier to manage and handle artifacts in a single place instead of copying and moving them manually. Nexus Pro simplifies access to repositories for your developers. For example, you can "hide" several public repository URLs behind the repository manager URL. This eliminates the need to configure repository access using configuration settings or project POMs. Nexus Pro also can be configured to download remote indexes and index the local artifacts for faster access from inside your network. You can also share your components externally using external repositories like the Central. Repository. Maven and Nexus Pro provide the capabilities necessary to deploy your artifacts to the Central Repository -- you can leverage the same approach that open source projects teams like Struts, Log4j, etc., use to publish your artifacts to a broad audience.

Build applications without tracking down and managing the source files - building a large application that consists of many components is much easier to do when you can assemble the application from binary components vs. building the application using the source code for each of the components. This is a significant time-savings because developers don’t need to find and manage the source files and the build process is simplified since the components don't have to be recompiled from the original source.

Easily organize your components - Nexus Pro helps you effectively manage multiple repositories. Even if you have a small development team, you will benefit from the ability to use different repositories. For example:

- **Snapshot and Release** - Nexus Pro provides the ability to manage Snapshot releases that help you manage components that are currently being developed. In addition, Nexus Pro provides staging and promotion capabilities that help you manage the entire dev to production process. Nexus also supports Release repositories that host fixed versions of artifacts so that you can support environments that require a consistent (non-changing) set of components.
- **Manage internal & external components** - With Nexus, you can manage your internal and external components separately. Internal repositories can be used to hold artifacts that your project or projects produce. External repositories can be used to host components from external sources. These can be OSS components from external repositories like the Central Repository, Codehouse, etc.or "external" components from other project teams in your organization, 3rd party commercial artifacts, etc.
- **Support multiple, related project teams** - With Nexus Pro, you can segregate artifacts into different repositories to meet the needs of different projects/applications and to coordinate efforts between related teams. Development teams are often organized into different supporting roles, such as GUI team, data access, external interfaces, etc. Nexus provides the ability to manage snapshots and release repositories. For example, snapshots can be used to share and manage components within one of these teams while release repositories can be used to make artifacts available to the related teams.
- **Support collaborative cross-company efforts** - The same advantages that apply to multiple project teams within the

same organization can also help manage efforts across multiple organizations. For example, when outsourcing or cooperating with multiple external companies or development teams on a larger project, a repository manager can be used to provide the components to everyone.

Easily manage access control via permission based sharing - You can use the Nexus Pro security subsystem to manage access to components using fine-grained access control. You can use this approach to partition repositories by job role. The Apache repository is a great example of partitioning in action. We use a single pair of release and snapshot repositories yet each project is only able to deploy the artifacts "owned" by their project. In addition to granular access control, Nexus Pro provides a host of security features, including LDAP support, SSO, SSL delivery, etc.

Easily manage multiple repository types, build tools, CI environments and programming languages - Many organizations have a heterogeneous environment. We often think about organizations that use multiple programming languages, multiple databases, multiple operating systems, etc. The same goes for build tools, Continuous Integration technologies, repository formats, etc. Nexus provides a single repository management infrastructure to support multiple build tools, repository formats and programming languages. This saves time and effort since you can use a single infrastructure to manage projects relating to Java and .NET; different repository formats such as Maven, Gradle, OSGI, Nuget; different CI tools like Hudson, Jenkins; and different build tools like Maven, Ant/Ivy, etc.

Strengthen / simplify your Continuous Integration / Deployment efforts - Nexus Pro promotion and staging is a natural complement to CI efforts. If your application is being continuously built and deployed using a tool like Hudson, a developer can checkout a specific module from a large multi-module build and not have to constantly deal with the entire source tree at any given time. This allows your software development effort to scale efficiently. If every developer working on a complex enterprise application needs to checkout the entire source tree every time he or she needs to make a simple change to a small component, building the application becomes a burdensome bottleneck. A slow enterprise build prevents the quick turnaround or quick feedback loop that helps your developers maintain focus during a development cycle. Once you are building with Maven, sharing binary artifacts with Nexus, continuously testing and deploying with Hudson, and generating reports and metrics with tools like Sonar, your entire organization gains a collaborative "central nervous system" that enables a more agile approach to software development.

> ***Nexus Pro supports more than Maven.*** *This may come as a surprise since the founders of Nexus also founded Maven and since Maven is also a popular repository format driven by Nexus usage, the two technologies are closely related. However, Nexus Pro supports a variety of build tools, repository formats and runtime environments.*

Speak with the Pros

Need help? Check out the next Nexus Live (https://plus.google.com/u/0/b/117596333621717490325/events/cgec45clnh4sambu7oks5jfp27s)
Discuss in the Nexus Community HipChat Room (https://links.sonatype.com/products/nexus/community-chat)

Helpful Resources

Configure Maven to Deploy Nexus Pro (https://support.sonatype.com/entries/21283268-Configure-a-Maven-

Build-to-Publish-to-Nexus)
Configuring Maven to a Single Nexus Group (https://support.sonatype.com/entries/20943003-Configure-Maven-to-Download-from-Nexus)

Terms of Service (/usage/terms-of-service)   Privacy Policy (/usage/privacy-policy)