Sonatype

PRODUCT OVERVIEW

NEXUS FOR NUGET-BASED DEVELOPMENT

An Introduction to Component Management

If you are developing on the Microsoft platform including .NET, you are either already using NuGet or you need to look at it. NuGet is a package manager providing tools that allow you to produce as well as consume packages. A large number of packages from Microsoft, such as Microsoft ASP.NET MVC or EntityFramework as well as the community created libraries such as NHibernate, jQuery or Bootstrap are available on the central package repository of the NuGet platform called NuGet Gallery.

By using NuGet you gain access to great Visual Studio integration as well as command line tools allowing you to install and use these libraries and many more with ease. With NuGet packages, you can build upon the open source features provided to create great applications without reinventing the wheel.

If you would like more information about development with NuGet, check out their documentation to get started.

Common Constraints

Once you are using NuGet and enjoy its benefits you will run into a some common development constraints (but don't worry, we're here to help). You will find:

Slower builds across dev teams

Each developer in your team as well as each build running on your continuous integration server nodes can trigger downloads from the NuGet Gallery. This, in turn, slows down builds. The reliance on external downloads might lead you to look for a way to avoid these repeated downloads over slow, external network connections relying on external servers.

Inefficient storage and sharing

Once the packages used are downloaded, you will want to decide where to store them. A common approach might be to check in the packages into the version control system, to share them via a fileshare, or to share them via some custom running web server. While these approaches work, you will find they are sometimes difficult to scale well or create problems with version updates, branches, etc.

Challenging collaboration

When working with other teams within or even outside your organization, creating your own proprietary packag-

es and sharing them via NuGet is a great way to divide responsibilities and accelerate build cycles for large complex applications. However, you will need to identify a good, common location to store and share these packages.

Coordinating configurations

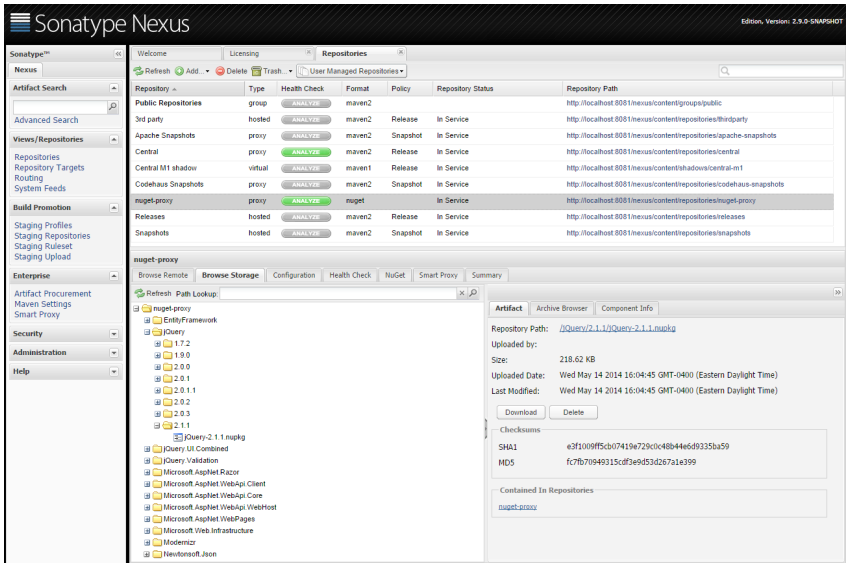
When using a number of open source packages from the NuGet Gallery and other repositories, all developer setups need to be configured to have access to all these packages. Then those setups need to be kept up to date. In larger development teams, this effort becomes increasingly complex to manage.

A Trusted Solution

All the problems found with using NuGet are minor compared to the benefits of using a package manager.

A category of a dedicated server software managing packages, also known as components, in repositories has emerged and Sonatype Nexus is the world's leading component manager. Today, over 40,000 organizations trust Nexus to enable faster, more reliable builds.

You can think of Nexus as a version control software server dedicated to managing binaries rather than source code. Nexus is available as open source and commercial software -- both of which include support for NuGet repositories.



Today, over 40,000 organizations trust Nexus to enable faster, more reliable builds.

A simple installation and configuration of Nexus allows you to avoid the problems mentioned above, while you gain access to further advantages (e.g., visibility to license types, alerts to published security risks associated with NuGet packages).

Getting Started

You can get started using Nexus in a couple of minutes following a number of easy steps:

- Install Nexus on a Windows or Linux server, or for initial tests even just on your own development machine
- Configure a proxy repository for the NuGet Gallery
- Create a hosted repository to store you internally created and maybe proprietary NuGet packages
- Create a repository group that bundles the two repositories you just created and exposes them as one unit
- Configure your VisualStudio IDE and your command line tools to get all packages from the NuGet repository group provided by Nexus

This initial setup will allow you to avoid repeated downloads from the NuGet Gallery since Nexus will proxy the components after the first download. You can host your own packages and potentially third party packages on your Nexus server, then provide access to them for your teams and CI servers.

Optionally, you can create proxy repositories to access other NuGet repositories of external companies or organizations. You can then add them to the repository group. With Nexus, you can also create additional hosted repositories (e.g., for third party packages you would like to add to the group). As soon as these repositories are added to the group, the packages are available without the need for any further configuration on the client side of your developer and CI server machines.

After enjoying these basic Nexus features, you can begin to explore its other features. For example, you can create packages of your application, store them in Nexus and retrieve them via HTTP or the REST API to deploy into your QA or production environments.

Further Benefits

You can get started using Nexus in a couple of minutes following a number of easy steps:

- A user interface to manage the repositories and packages
- Control of the user access via a security setup that can be integrated with Active Directory or other LDAP backends
- A search interface for finding available packages
- Package details such as known security vulnerability and license data

Nexus also supports other repository formats such as Maven, YUM, OBR or NPM that might be useful in your organization allowing you to avoid running several applications to solve similar problems.

Summary

If you develop software, Nexus can help you share your NuGet packages with other developers and end users. While the NuGet Gallery has always served as a great convenience for users of Visual Studio and other common development tools, maintaining your own repositories is recommended to ensure stability within builds across your organization. Nexus greatly simplifies the maintenance of your own internal package management and access to external package managers. With Nexus you can completely control access to, and deployment of, every NuGet package in your organization from a single location.

Get started with Nexus today!

Sonatype Inc. • 8161 Maple Lawn Blvd, Suite 250 • Fulton, MD 20759 • 1.877.866.2836 • www.sonatype.com