

NuGet

On this page:

- [Integration Capabilities](#)
- [Typical Usage Scenarios](#)
- [Installing NuGet to TeamCity agents](#)
- [NuGet Packages Cache Clean-up on Agents](#)
- [Using TeamCity as NuGet Server](#)
- [Symbol Packages](#)
- [Proxy Configuration](#)

Integration Capabilities

TeamCity integrates with [NuGet](#) package manager and when [NuGet is installed](#) provides the following capabilities:

- [NuGet feed](#) based on the builds' published artifacts
- A set of NuGet runners to be used in builds on Windows OS; and since TeamCity 2017.1 on Linux and Mac OSX when [Mono](#) is installed on the agent. Only NuGet 3.2+ on Mono 4.4.2+ is supported.
 - the [NuGet Installer](#) build runner, which installs and updates NuGet packages
 - the [NuGet Pack](#) build runner, which builds NuGet packages
 - the [NuGet Publish](#) build runner, which publishes packages to a feed of your choice
- The [NuGet Dependency Trigger](#), which allows triggering builds on NuGet feed updates.



Supported Operating Systems

NuGet build runners are supported on build agents running Windows OS by default. Since TeamCity 2017.1 Linux and Mac OSX are supported when [Mono](#) is installed on the agent (NuGet command-line executable for Windows version 3.2 and later is strongly recommended with Mono).



- NuGet build runners require an appropriate version on .NET Framework installed on the agent machine depending on the NuGet.exe version used: NuGet 2.8.6+ requires .NET 4.5+, earlier NuGet versions require .NET 4.0.
- To use packages from an authenticated feed, see the [NuGet Feed Credentials](#) build feature.

Typical Usage Scenarios

- To install packages from a public feed, add the [NuGet Installer](#) build step.
- To create a package and publish it to a public feed, add the [NuGet Pack](#) and [NuGet Publish](#) build steps.
- To create a package and publish it to the internal TeamCity NuGet Server, enable TeamCity as a NuGet Server (see the [section](#) below), use the [NuGet Pack](#) build step and [NuGet Publish](#) build steps.
- To trigger a new build when a NuGet package is updated, use [NuGet Dependency Trigger](#).

Installing NuGet to TeamCity agents

The NuGet trigger and the NuGet-related build runners require the NuGet command line binary configured on the server. They are automatically distributed to agents once configured. Several versions can be installed and a version of your choice can be set as the default one.

To install NuGet.exe on TeamCity:

1. Go to the Administration | Tools tab.
2. Click Install tool and select NuGet.exe.
3. Select whether you want to download (default) NuGet from the public feed or upload your own NuGet package containing `NuGet.exe`.
 - a. If the Download radio button is chosen, select the NuGet version to install on all build agents.



It is recommended to use release versions of NuGet.

- b. If the Upload radio button is selected, choose your own NuGet package.
4. Specify whether this NuGet version will be default using the related check-box.

5. Click Add to install the selected NuGet version.

 Installing NuGet on agents results in agents upgrade.

NuGet Packages Cache Clean-up on Agents

NuGet uses several local caches to avoid downloading packages that are already installed, and to provide offline support.

Since TeamCity 2017.2, if an agent is running out of the space, TeamCity will try to clean NuGet packages cache on the agent.

The caches in the following directories will be cleaned:

- %NUGET_PACKAGES% environment variable (must be an absolute path)
- %LOCALAPPDATA%\NuGet\Cache
- %LOCALAPPDATA%\NuGet\v3-cache
- %user.home%.nuget\packages

Using TeamCity as NuGet Server

If for some reason you don't want to publish packages to public feed, e.g. you're producing packages that are intended to be used internally; you can use TeamCity as a NuGet Server instead of setting up your own repository.

 TeamCity running on any of the supported operating systems (Windows, Linux, Mac OS X) can be used as a NuGet Server.

Since TeamCity 9.0 built-in NuGet feed supports API v2 by default.

TeamCity NuGet feed users are recommended to upgrade to TeamCity 10.0.2 and later, as 10.0.2 provides improved compatibility with NuGet.

To start using TeamCity as a NuGet Server, go to the Administration | NuGet Feed page (available to server system administrators), and click Enable. Two different links will be displayed on the page: for public (with `guestAuth` prefix) and private (with `httpAuth` prefix) feed. If Public Url is not available, you need to enable the [Guest user login](#) in TeamCity on the Administration | Global Settings page.

When the NuGet Server is enabled, all NuGet published packages will be indexed and will appear in NuGet feed. The feed will include the packages from the build configurations where the currently authenticated user has permission to view build artifacts ("View project" permission).

When you have TeamCity NuGet server enabled:

- You can publish packages to the feed either as build artifacts of the [NuGet Pack](#) build step (using the Publish created packages to build artifacts checkbox, packages will be indexed asynchronously) or via the [NuGet Publish](#) build step (since TeamCity 2017.1 packages will be indexed synchronously)
- You can add TeamCity NuGet server to your repositories in Visual Studio to avoid having to type in long URLs each time you want to read from a specific package repository (add NuGet repository and specify the public URL provided by TeamCity when enabling NuGet server).
- The packages available in the feed are bound to the builds' artifacts: they are removed from the feed when the artifacts of the build which produced them are [cleaned up](#).
- You do not need to specify credentials for the internal NuGet feed access; if [NuGet Feed Credentials](#) are specified, they are ignored.

If the build artifacts are changed under TeamCity Data Directory manually, you need to instruct TeamCity to reindex NuGet feed. For that, click "reset" link for "buildsMetadata" under Administration > Diagnostic > Caches.

 Internet Explorer settings may need to be set to trust the TeamCity Server when working in a mixed authentication environment.

For a step-by-step example of NuGet setup see blog post: [Setting up TeamCity as a native NuGet Server](#).

Symbol Packages

Publishing of NuGet [symbol packages](#) to the internal TeamCity feed may cause issues when using an external source server. See the [corresponding issue](#) in our public tracker.

Proxy Configuration

NuGet command line client supports proxy server configuration via NuGet.config file parameters or environment variables. See [NuGet documentation](#) for more details.

See also:

[Administrator's Guide: NuGet Installer](#) | [NuGet Publish](#) | [NuGet Pack](#) | [NuGet Dependency Trigger](#)