What is NuGetForUnity?

NuGetForUnity is a NuGet client built from scratch to run inside the Unity Editor. NuGet is a package management system which makes it easy to create packages that are distributed on a server and consumed by users. NuGet supports sematic versioning for packages as well as dependencies on other packages.

You can learn more about NuGet here: nuget.org

NuGetForUnity provides a visual editor window to see available packages on the server, see installed packages, and see available package updates. A visual interface is also provided to create and edit *.nuspec* files in order to define and publish your own NuGet packages from within Unity.



How do I install NuGetForUnity?

Install the provided Unity package into your Unity project. Located here.

How do I use NuGetForUnity?

To launch, select NuGet
→ Manage NuGet Packages

🚭 Unity Personal (64bit) - Untitled - Packager - Web Player <DX11>

File Edit Assets	GameObject Component	NuGet Asset	Store Tools	Window	Help
🖑 🕂 🖱	Manage NuGet Packages	45			
f≡ Hierarchy	Restore Packages		# Scene	CG	ame
Create * Q*All	Show Dependency Tree	D^{-}	Shaded	-	2D 🗦
Main Camera	Version 0.0.12				

After several seconds (it can take some time to query the server for packages), you should see a window like this:

NuGet		= × *=
Online	Installed	Updates
Show All Versions		Refresh
Show Prerelease		
Search		Search
4 Json.NET [10.0.3]		Update [10.0.1]
Json.NET is a popular high-performance JSON framewo	ork for .NET	
View License		
ို္ဝူNUnit [3.9.0]		Install
NUnit features a fluent assert syntax, parameterized,	generic and theory tests and is user-extensible.	
This package includes the NUnit 3 framework assembl third-party runner that supports NUnit 3 in order to exe	y, which is referenced by your tests. You will need to install ecute tests. Runners intended for use with NUnit 2.x will not	version 3 of the nunit3-console program or a trun NUnit 3 tests correctly.
Supported platforms: NET 2.0+ NET Standard 1.3 and 1.6 NET Core		
Release Notes:		
This package includes the NUnit 3 framework assembl third-party runner that supports NUnit 3 in order to ex- View License	y, which is referenced by your tests. You will need to install ecute tests. Runners intended for use with NUnit 2.x will not	version 3 of the nunit3-console program or a run NUnit 3 tests correctly.
EntityFramework [6.2.0]		Install
Entity Framework is Microsoft's recommended data acc	ess technology for new applications.	
View License		
ojQuery [3.2.1]		Install
jQuery is a new kind of JavaScript Library. jQuery is a fast and concise JavaScript Library th development. jQuery is designed to change the way th NOTE: This package is maintained on behalf of t	at simplifies HTML document traversing, event handling, ar nat you write JavaScript. the library owners by the NuGet Community Packages proje	vimating, and Ajax interactions for rapid web ct at http://nugetpackages.codeplex.com/
View License		
HtmlAgilityPack [1.6.4]		Install
This is an agile HTML parser that builds a read/write D	OM and supports plain XPATH or XSLT (you actually don't H	IAVE to understand XPATH nor XSLT to use it, don't

The **Online** tab shows the packages available on the NuGet server.

Enable **Show All Versions** to list all old versions of a package (doesn't work with nuget.org). Disable **Show All Versions** to only show the latest version of a package.

Enable **Show Prelease** to list prerelease versions of packages (alpha, beta, release candidate, etc). Disable **Show Prerelease** to only show stable releases.

Type a search term in the **Search** box to filter what is displayed.

Press the **Refresh** button to refresh the window with the latest query settings. (Useful after pushing a new package to the server and wanting to see it without closing and reopening the window.)

The name of the package, the version of the package (in square brakets), and a description are displayed.

Click the View License to open the license in a web browser.

Click the **Install** to install the package. Note: If the package is already installed an **Uninstall** button will be displayed which lets you uninstall the package.

The Installed tabs shows the packages already installed in the current Unity project.

NuGet		□ × *=		
Online	Installed	Updates		
Search		Search		
Json.NET [10.0.1]		Uninstall		
Json.NET is a popular high-performance JSON framework for .NET				

View License

Click the **Uninstall** button to uninstall the package.

The Updates tab shows the packages currently installed that have updates available on the server.

 NuGet
 Installed

 Online
 Installed

 Show All Versions
 Install All Updates

 Show Freedeace
 Search

 Search
 Search

 Json.NET [10.0.3]
 Update [10.0.1]

The version in brackets on the left is the new version number. The version in brackets in the **Update** button is the currently installed version.

Click the Update button to uninstall the current package and install the new package.

How does NuGetForUnity work?

NuGetForUnity loads the *NuGet.config* file in the Unity project (automatically created if there isn't already one) in order to determine the server it should pull packages down from and push packages up to. By default, this server is set to the nuget.org package source.

The default NuGet.config file:

View License

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <packageSources>
        <add key="NuGet" value="http://www.nuget.org/api/v2/" />
        </packageSource>
        <adtivePackageSource>
        <add key="NuGet" value="http://www.nuget.org/api/v2/" />
        </activePackageSource>
        <config>
        <add key="repositoryPath" value="./Packages" />
        <add key="DefaultPushSource" value="http://www.nuget.org/api/v2/" />
        </config>
        </configuration>
        </configuration
```

You can change this to any other NuGet server (such as NuGet.Server or ProGet - see below). The NuGet \rightarrow Reload NuGet.config menu item is useful if you are editing the *NuGet.config* file.

See more information about NuGet.config files here: https://docs.nuget.org/consume/nuget-config-settings

🚭 Unity Personal (64bit) - Untitled - Packager - Web Player <DX11>



NuGetForUnity installs packages into the local repository path defined in the *NuGet.config* file (repositoryPath). By default, this is set to the Assets/Packages folder. In the *NuGet.config* file, this can either be a full path, or it can be a relative path based on the project's Assets folder. Note: You'll probably want your Packages folder to be ignored by your version control software to prevent NuGet packages from being versioned in your repository.

When a package is installed, the *packages.config* file in the project is automatically updated with the specific package information, as well as all of the dependencies that are also installed. This allows for the packages to be restored from scratch at any point. The Restore operation is automatically run every time the project is opened or the code is recompiled in the project. It can be run manually by selecting the **NuGet** \rightarrow **Restore Packages** menu item.

🚭 Unity Personal (64bit) - Untitled - Packager - Web Player <DX11>



Note: Depending on the size and number of packages you need to isntalled, the Restore operation could take a *long* time, so please be patient. If it appears the Unity isn't launching or responding, wait a few more minutes before attempting to kill the process.

If you are interested in the process NuGetForUnity follows or you are trying to debug an issue, you can force NuGetForUnity to use verbose logging to output an increased amount of data to the Unity console. Add the line <add key="verbose" value="true" /> to the <config> element in the *NuGet.config* file. You can disable verbose logging by either setting the value to false or completely deleting the line.

The .nupkg files downloaded from the NuGet server are cached locally in the current user's Application Data folder. (C:\Users\[username]\AppData\Local\NuGet\Cache). Packages previously installed are installed via the cache folder instead of downloading it from the server again.

How do I create my own NuGet packages from within Unity?

First, you'll need to create a *.nuspec* file that defines your package. In your Project window, right click where you want the *.nuspec* file to go and select NuGet \rightarrow Create Nuspec File.

	Create Show in Explorer Open Delete	>
	Import New Asset Import Package Export Package Find References In Scene Select Dependencies	>
	Refresh Reimport	Ctrl+R
	Reimport All	
	Run API Updater	
	Open C# Project	
Treate Nuspec File Open Nuspec Editor	NuGet	>

Select the new .nuspec file and you should see something like this:

MyPackage	*=
ID	MyPackage
Version	0.0.1
Authors	Your Name
Owners	Your Name
License URL	http://your_license_url_here
Project URL	http://your_project_url_here
Icon URL	https://www.nuget.org/Content/Images/packageDefaultIcon-50x50.png
Require License	
Description	A description of what this packages is and does.
Release Notes	Notes for this specific release
Copyright	Copyright 2017
Tags	
▼ Dependencies	
	Automatically Fill Dependencies
	Add Dependency
L	Save MyPackage.nuspec
	Pack MyPackage.nupkg
API Key	
	Push to Server

Input the appropriate information for your package (ID, Version, Author, Description, etc). Be sure to include whatever dependencies are required by your package.

Press the **Pack** button to pack your package into a *.nupkg* file that is saved in the C:\Users\ [username]\AppData\Local\NuGet\Cache folder.

Press the **Push** button to push your package up to the server. Be sure to set the correct API Key that give you permission to push to the server (if you server is configured to use one).

How do I create my own NuGet server to host NuGet packages?

You can use NuGet.Server, NuGet Gallery, ProGet, etc to create your own NuGet server.

Alternatively, you can use a "local feed" which is just a folder on your hard-drive or a network share.

Be sure to set the proper URL/path in the NuGet.config file and you should be good to go!