Creating a NuGet Package

Contents

[Approach 1 – Command line 3](#_Toc421523643)

[Approach 2 – NuGet Package Explorer 7](#_Toc421523644)

[Tips 9](#_Toc421523645)

[Readme/Help File 9](#_Toc421523646)

[XML Documentation 10](#_Toc421523647)

[Versioning 12](#_Toc421523648)

[Dependencies 14](#_Toc421523649)

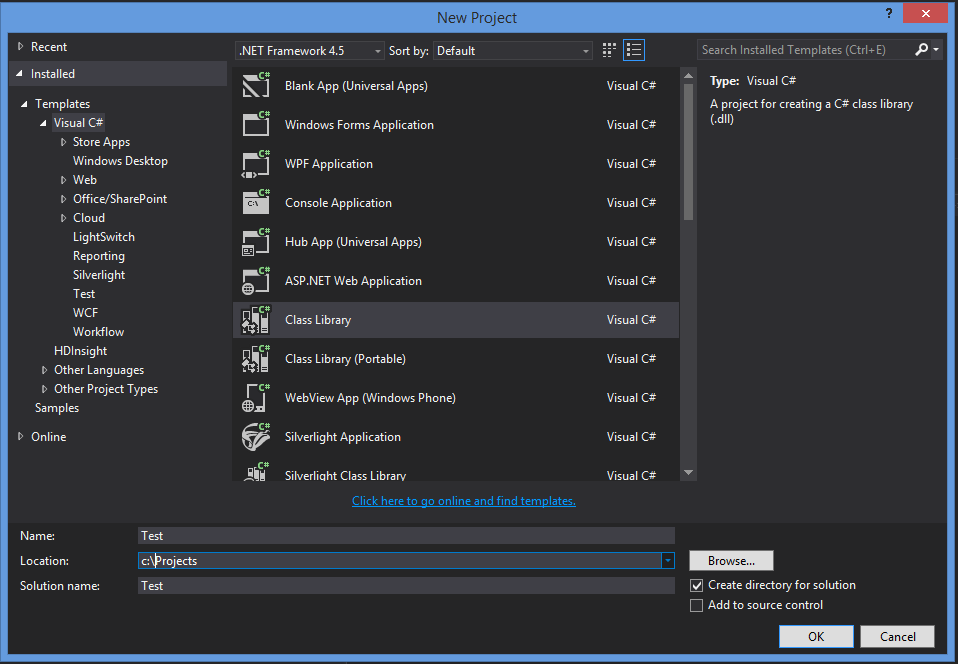
There’s two different ways to create a package;

1. Command line using NuGet.exe
2. NuGet Package Explorer – I would say this is the easier approach having done both

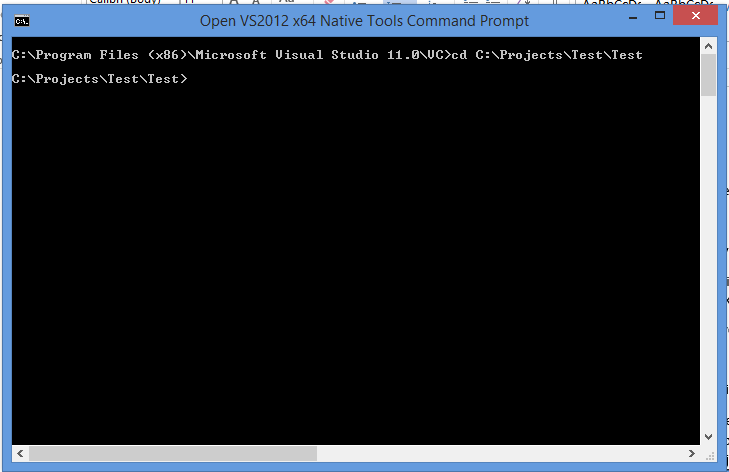
This article is going to cover both for completeness

# Approach 1 – Command line

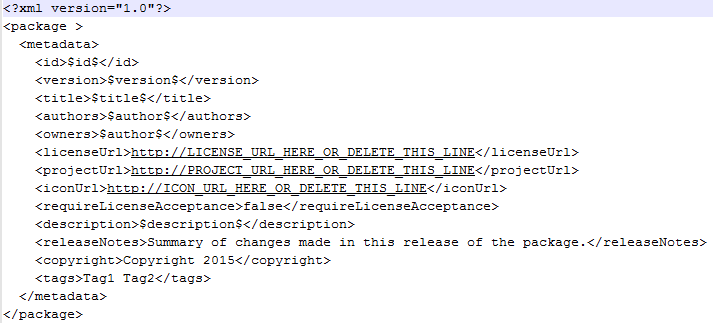
1. Download nuget.exe <https://nuget.org/nuget.exe>
2. Copy the file to somewhere you’ll remember the location of e.g. C:\NuGet
3. Create a new project of type Class Library called Test



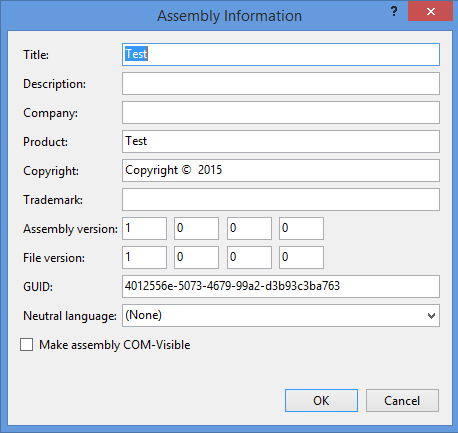
1. Build the solution
2. Open Command Prompt and navigate to the root of the project e.g.;



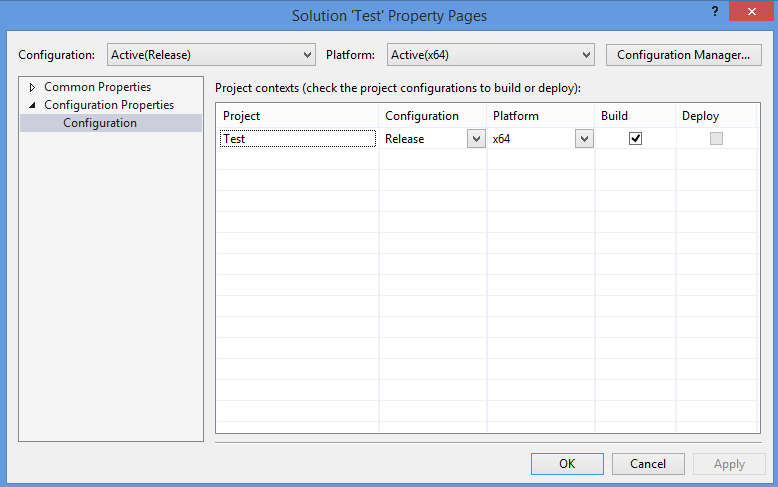
1. Next we need to create a nuspec file, this is a specification file for a nuget package
2. In the command prompt type the path to nuget.exe followed by spec; 
3. You can open this new file in something like Notepad ++ it is just an xml file with a different extension and will look similar to;



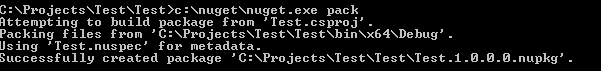
1. Set authors to the appropriate name and owners to 3Squared
2. Placeholders like $title$ get pulled from the project assembly information, to edit this right like on the project and click Properties then click on the Assembly Information button and you’ll see a popup like this;



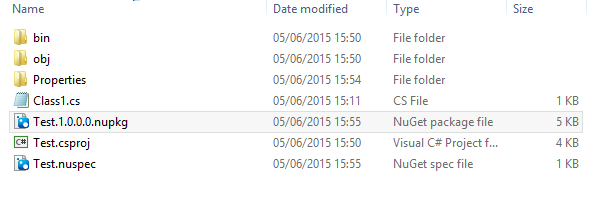
1. Add a description
2. Build the project – Important if you don’t the changes to assembly information will not be applied
3. Go back into command prompt and type the path to nuget.exe followed by pack
4. I found I had an error and had to set the platform on the solution;



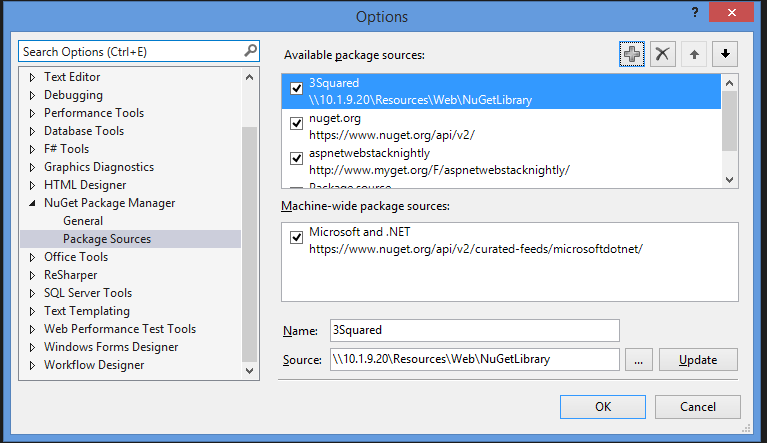
1. If an error occurs at command line retry the command but append the argument –verbosity detailed
2. Your package should have been created;



1. If your package has been created successfully you’ll be able to see it in Windows Explorer;



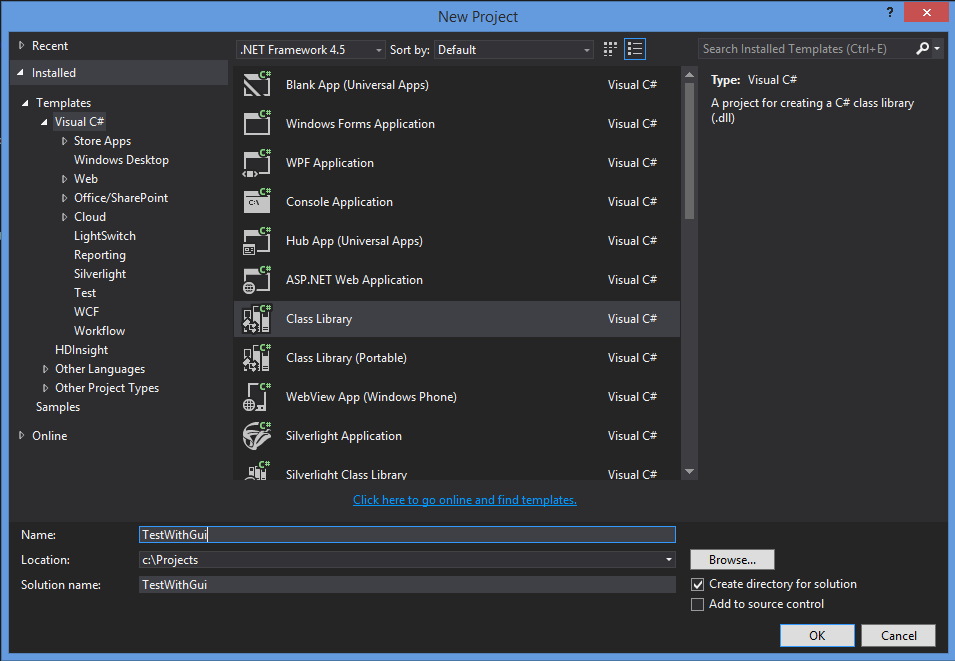
1. You can now create a local NuGet package source and consume your package or copy it to the 3Squared library for use within the team;



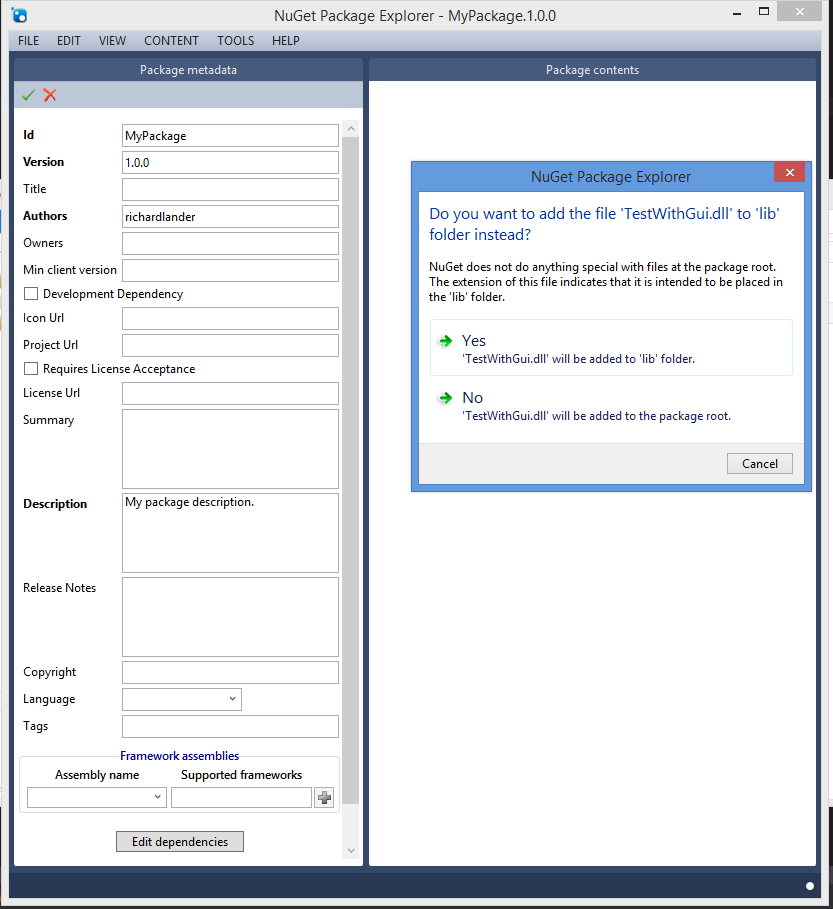
Useful link: <https://docs.nuget.org/create/creating-and-publishing-a-package>

# Approach 2 – NuGet Package Explorer

1. Create a new Project;



1. Install the application from <http://npe.codeplex.com/>
2. Open the application
3. Select “Create a new package”
4. Select Edit > Edit Metadata
5. Set properties as appropriate e.g. Id, description etc
6. Build your project
7. View in File Explorer and drag in the dll you would like to package up into the Package Contents folder of the application;



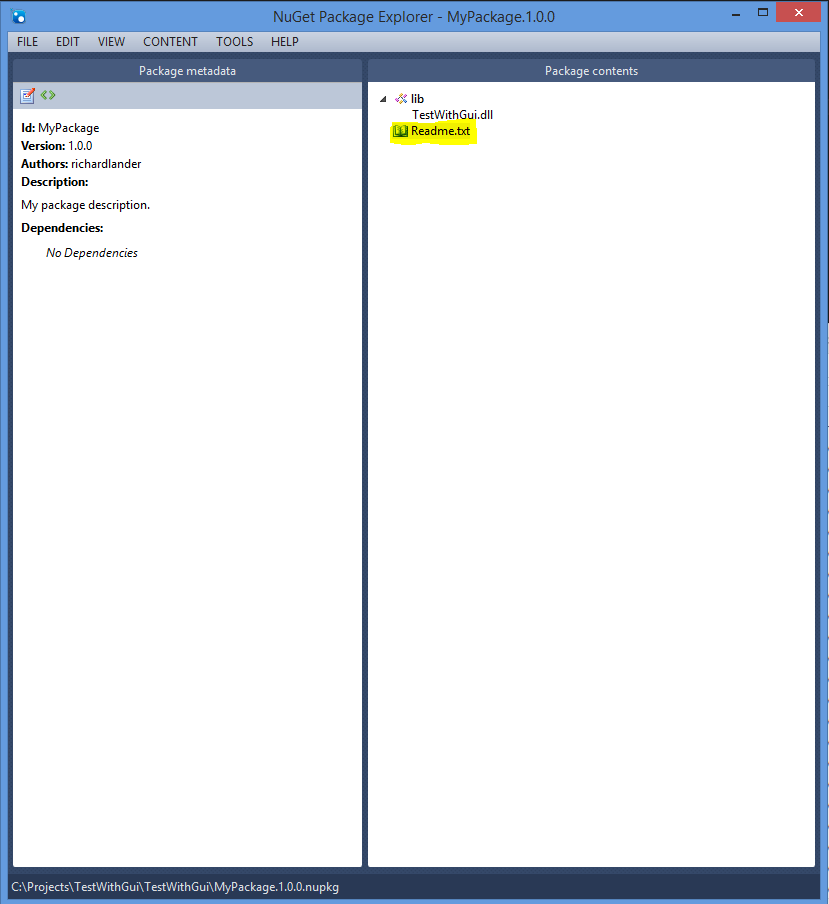
1. Click Yes
2. Save – that’s it, much easier from the command line

Useful link: <https://docs.nuget.org/Create/using-a-gui-to-build-packages>

# Tips

## Readme/Help File

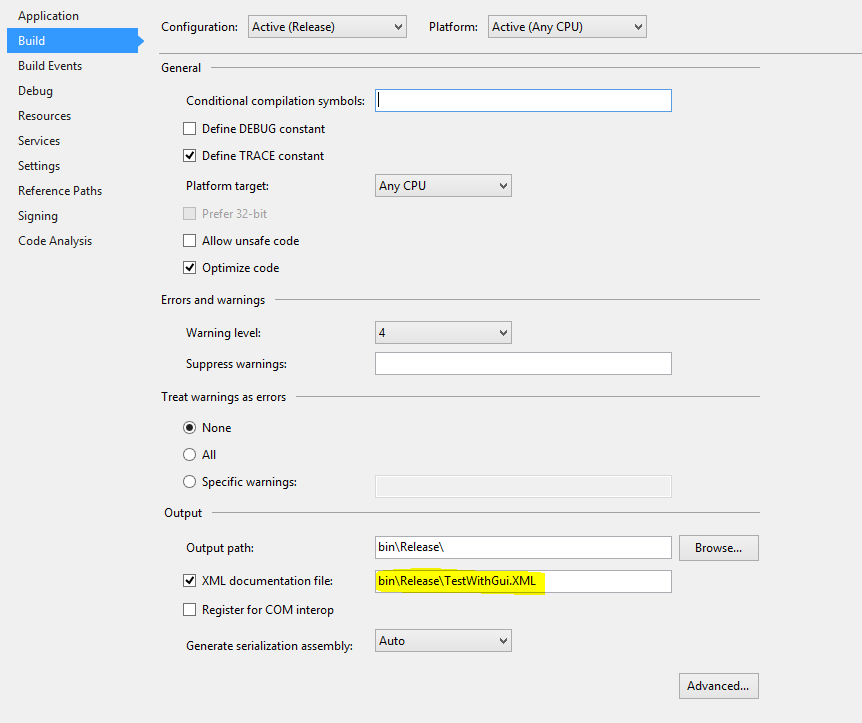
* Add a Readme.txt to the root of your NuGet package containing all the instructions to use your NuGet package;

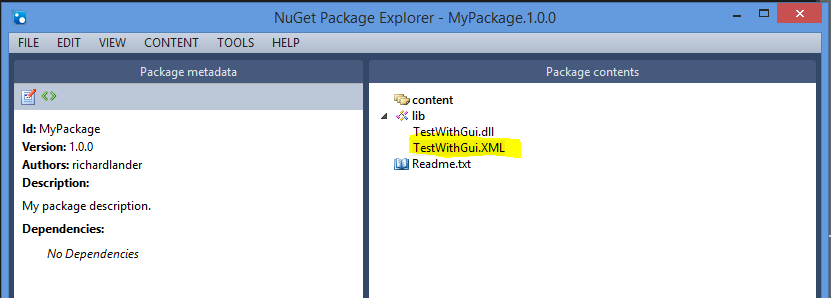


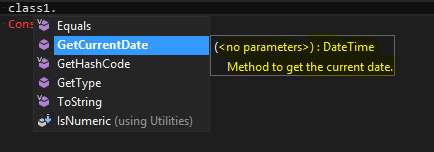
When the NuGet package is added to any application the readme file will automatically be opened displaying the instructions on how to use the package

## XML Documentation

* To include the xml documentation ensure XML documentation file is selected in the build properties;

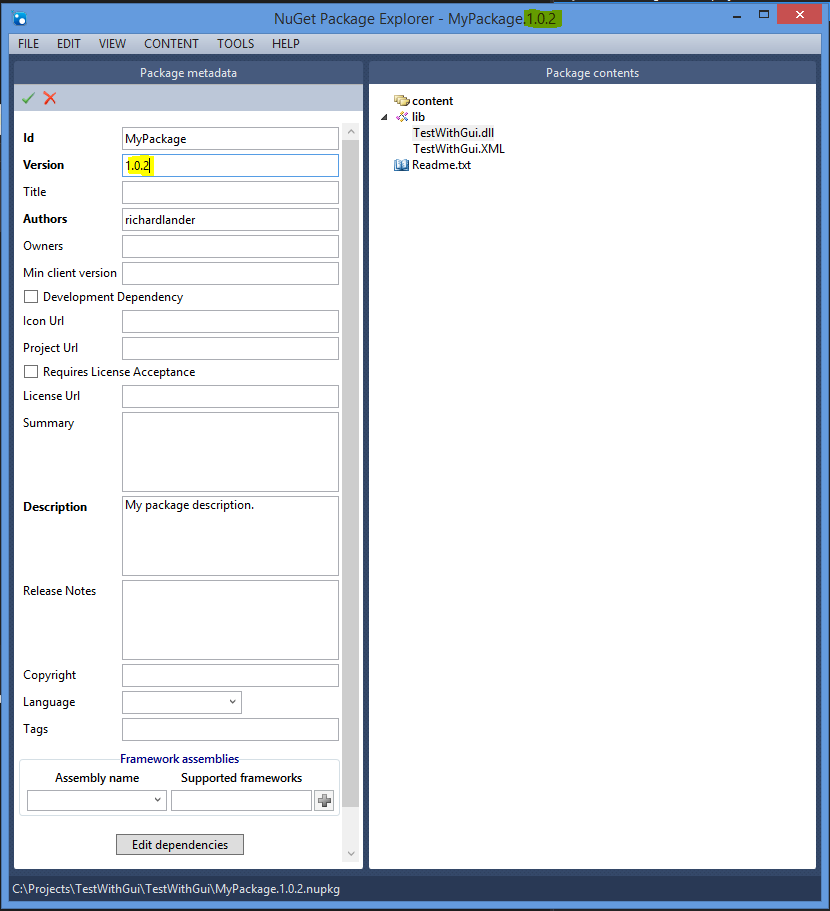


* Drag in the xml file; 
* XML documentation will now be available from the NuGet package;

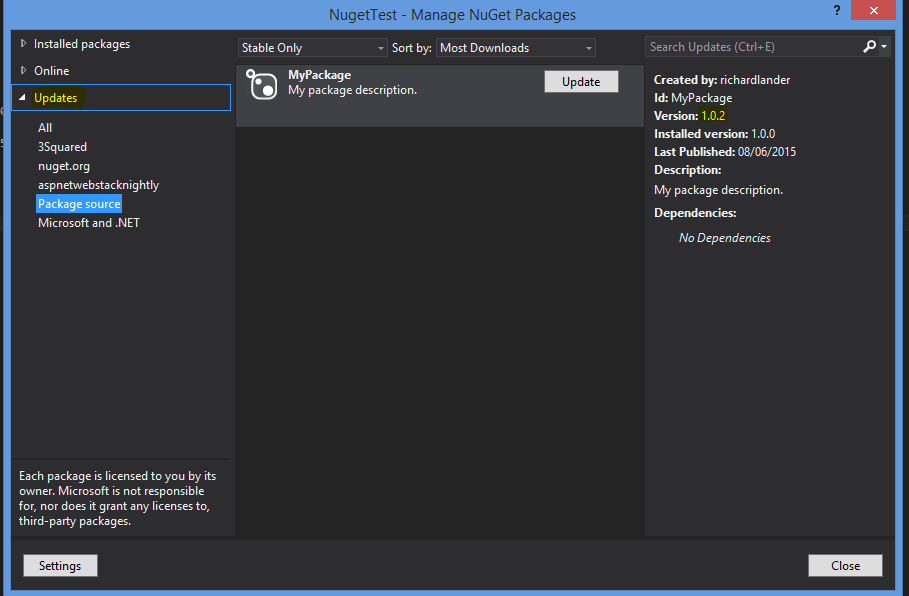


## Versioning

* Increment the version number of the metadata and save the package

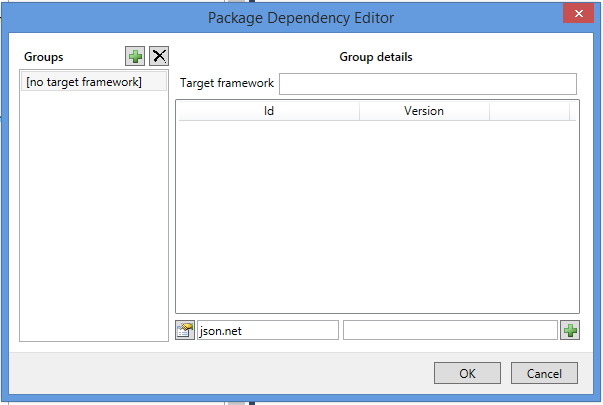


* Add release notes which include details of the changes
* Copy the new Nuget package over to the source
* If you go to Manage NuGet packages now and click updates you should see the update is available

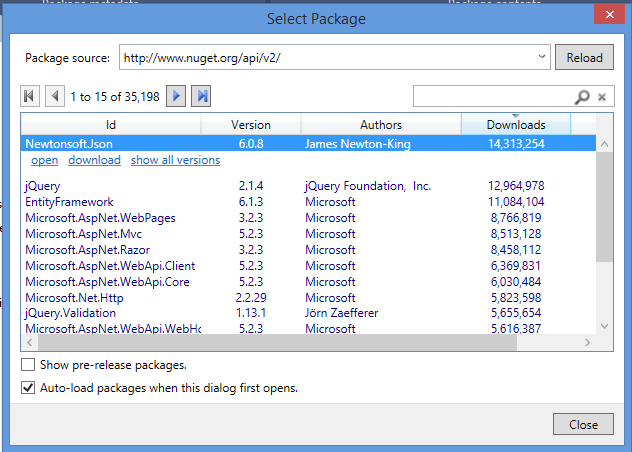


## Dependencies

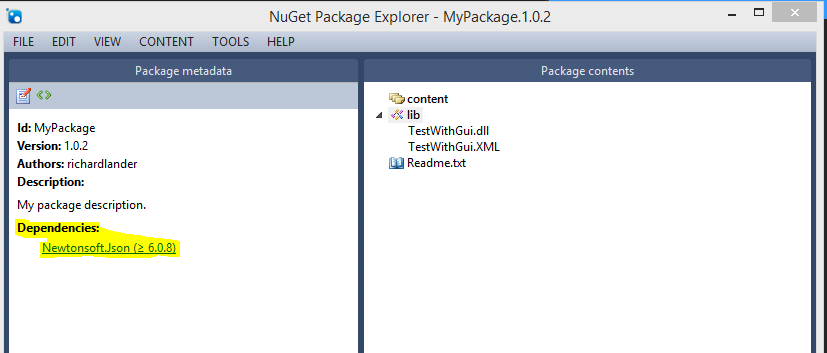
* Click edit dependencies
* Type the name into the text box



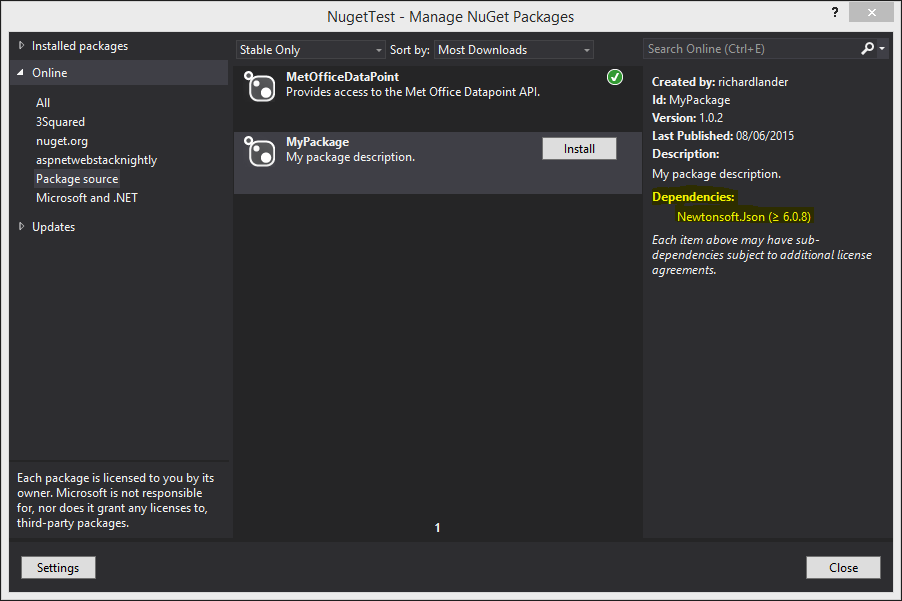
* Click the select button
* Find your dependency;



* Click Open to add it then ok
* Save your pack and you should now see the dependency;



* It will also be visible within Manage NuGetPackages;



* If your using the command line tool and dependencies are not included you can use the  [-IncludeReferencedProjects](https://docs.nuget.org/Consume/Command-Line-Reference#Pack-Command-Options) option