**Create NuGet Package from an Assembly DLL**

**1) Download Nuget.exe**

1. Visit [nuget.org/downloads](https://nuget.org/downloads) and select NuGet 3.3 or higher (2.8.6 is not compatible with Mono). The latest version is always recommended, and 4.1.0+ is required to publish packages to nuget.org.
2. Each download is the nuget.exe file directly. Instruct your browser to save the file to a folder of your choice. I usually place it in C:\Users\<user-name>\source. The file is *not* an installer; you won't see anything if you run it directly from the browser.

**2) Generate Nuspec File from DLL**

In the simple case of creating a package from an assembly. Within the command prompt, navigate to the directory where the DLL is.



You can generate a .nuspec file from the metadata in the assembly using the spec command. Point to where you placed the nuget.exe earlier, then call spec and pass the name of the DLL to generate from.



**3) Edit the Manifest**

Right click on the .nuspec file that was generated and edit. You must edit the following fields before proceeding. See [.nuspec file reference - optional metadata elements](https://docs.microsoft.com/en-us/nuget/reference/nuspec#optional-metadata-elements) for a description of how these are used.

* licenseUrl
* projectUrl
* iconUrl
* releaseNotes
* tags

For packages built for public consumption, pay special attention to the **Tags** property, as tags help others find your package on sources like nuget.org and understand what it does.

You can also add any other elements to the manifest at this time, as described on [.nuspec file reference](https://docs.microsoft.com/en-us/nuget/reference/nuspec).

Save the file before proceeding.

Navigating to the project folder and doing a ‘dotnet pack SampleLibrary.csproj’ will also create an entire nupkg the way it thinks it should.

**4) Create the NuGet Package**

After the manifest (.nuspec) file is complete and correct, the nuget.exe can pack it into a .nupkg package.



If you get warning, such as DLLs not being included in a “lib” folder, you can manually correct them at this point. Rename the .nupkg file to instead have a .zip extension. Then you can extract the contents and make your adjustments. Then right-click on the folder, send to compressed (zipped) folder. Then rename that file to use the original .nupkg extension.

**5) Install NuGet Package**

You can publish your new NuGet package, which would make it available from nuget.org, but for now we will install it from an offline source.

1. Paste the .nupkg file into C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\
2. Then in the desired solution in Visual Studio, select Tools > NuGet Package Manager > Package Manager Console
3. Within the Package Manager Console pane, change the Package source to be “Microsoft Visual Studio Offline Packages” and then change the Default project to be the project that will reference the assemblies included in the NuGet package.
4. Run the Install-Package command with the id (from the manifest) of the NuGet package.

