

How To Add Release Notes to Nuget Package

These are the steps needed to include release notes in a Nuget package.

These are taken from this article:

[Writing a NuGet package release notes in an outside of a .csproj file.](#)

Same article here:

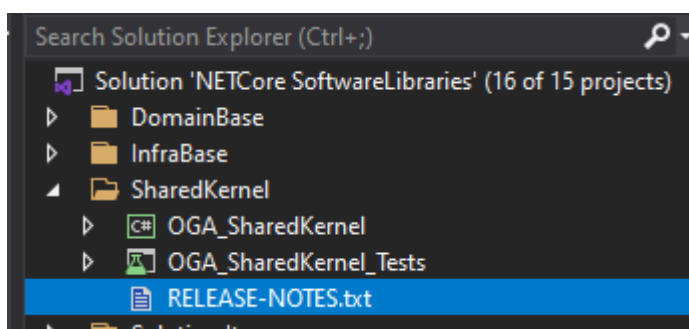
[Writing a NuGet package release notes in an outside of a .csproj file.](#)

Release Notes are kept in the VS solution with the project, but are not stored inside the project.

A target block is added to the csproj file so that MSBuild can include the release notes content into the nuget package.

How To

1. Create a text file in the VS Solution, called: `RELEASE-NOTES.txt`, or something more specific, if more than one release notes file is needed.
2. Place the release notes file outside the project node, but still in the VS Solution, such as this:



3. Add content to the release notes file, to annotate each version in descending order, like this:

```
1 Shared Kernel Release Notes
2
3 v1.0.3
4   Date: 20211215
5   Updates:
6     Added cStackTrace class to assembly.
7     This class provides the ability to create a stack trace with
8     Or, for stack frames that don't have a usable PDB for their
9     These latter two values maintain traceability to the source
10    The module token and IL Offset values can be used in a query
11    Download the "pdb2xml" converter tool from MSDN: http://www
12
13 v1.0.2
14   Date: 20211213
15   Updated to add Not Found exception.
16
17 v1.0.1
18   Date: 20211212
19   Updates:
20     No actual updates in this version.
21     The version was incremented to test the nuget pipeline.
22
23 v1.0.0
24   Date: 20211212
25   Updates:
26     No actual updates in this version.
27     This version was incremented to test the nuget pipeline.
28
29 v1.0.0-build2
30   Date: 20211212
31   Updates:
32     No actual updates in this version.
33     This version was created to test the nuget pipeline.
34
```

4. Add a comment in the Assembly Release Notes field, to point other developers to use the Release Notes.txt file.

This can be done by writing a comment in the PackageReleaseNotes MSBuild property, via the Project property editor GUI, or editing the csproj file.

We do this first, before we add the MSBuild target to the csproj file, to ensure the release notes property is set.

The screenshot shows the Project Property Editor GUI with the following fields and values:

- Repository type:
- Tags:
- Release notes:
- Assembly neutral language:
- Assembly version:
- Assembly file version:

5. Now, we need to create a MSBuild Target (like the below screenshot) in the csproj file so that MSBuild can pick up the release notes before the nuget package is built.

This action needs to be executed before the “GenerateNuspec” target is executed.

At the before executing of the “GenerateNuspec” target is a good entry point to prepare the package release notes.

This target reads the release notes file contents by “ReadLinesFromFile” MSBuild standard task, and outputs the contents into a MSBuild items which is named, ReleaseNoteLines.

Finally, MSBuild will build the PackageReleaseNotes MSBuild property from the ReleaseNoteLines MSBuild items that contain the contents of the release notes file.

By the way, MSBuild joins the items with a “;” separate by default.

In this case, we want to make PackageReleaseNotes MSBuild property with multi-lines, therefore we write the reference of ReleaseNoteLines MSBuild itens with “LF” (Line-Feed) separate explicitly, like this:

```
@(ReleaseNoteLines, '%0a')
```

('%0a' means LF code in an MSBuild script file.)

Here’s what the MSBuild target looks like:

```
<Target Name="PreparePackageReleaseNotesFromFile" BeforeTargets="GenerateNuspec">
  <ReadLinesFromFile File="./RELEASE-NOTES.txt" >
    <Output TaskParameter="Lines" ItemName="ReleaseNoteLines"/>
  </ReadLinesFromFile>
  <PropertyGroup>
    <PackageReleaseNotes>@(ReleaseNoteLines, '%0a')</PackageReleaseNotes>
  </PropertyGroup>
</Target>
```

This build target needs to be pasted into the csproj file, like this:

