

How to Create an Arcade Package

One of the goals of Arcade is to be a vehicle to provide code sharing. One of the ways to accomplish this is to pack each tool into its own NuGet package. This document assumes that you are adding a package called **ProjectOne** to Arcade and provides a list of things you should check while doing that.

Things to Check

- Package sources must go into the `src\` folder on the root of the repository.
- Inside the `src\` folder create a folder representing the project namespace: `src\Microsoft.DotNet.ProjectOne`.

- The main sources for the package must go into `src\` folder. That is:

```
src\Microsoft.DotNet.ProjectOne\src
```

- The sources for unit tests should go side-by-side with the main sources.

```
src\Microsoft.DotNet.ProjectOne\tests
```

- You should include a `README.md` file on the root of the package.
- Make sure the namespace prefix of the package follows the name of the project. For instance:

```
namespace Microsoft.DotNet.ProjectOne.Model
```

- Include the package on the `Arcade.sln` file. This way it will be compiled automatically when the solution is built.
- Make sure that library dependencies of the new package use the Arcade defined version of the library. The versions are defined in the `..\eng\Versions.props` file. For instance, if the project has a dependence on `Newtonsoft.Json` you must add it as follows:

```
<PackageReference Include="Newtonsoft.Json" Version="$(NewtonsoftJsonVersion)"
/>
```

If there is no version specified for a required library feel free to add a new property defining it.

- If your package produces binaries that need to be signed, make sure to mark it as true or check the `SignTool` documentation to see how to sign specific files.
- If the package needs to target a version of .NET Framework we recommend you to use the Arcade defined version, which is stored in the `$(NetFxTfm)` property. For instance:

```
<TargetFrameworks>$(NetFxTfm);netcoreapp2.0</TargetFrameworks>
```

- There is no requirement to create a separate `.nuspec` file for the package. The package information will be automatically extracted from the `.csproj` file.

Further References

- `Overview.md`
- `..\eng\Versions.props`

Was this helpful?  