

# How to have a Project Reference without referencing the actual binary

04/04/2015 • 2 minutes to read

Sometimes you want a project reference from project B to project A to indicate a build-time dependency, but you don't actually want assembly B to reference assembly A (maybe because it's a runtime-only dependency or loaded using reflection). Having a project reference is beneficial because you indicate to the build system that in order to have B fully built you also need to build (and possibly deploy) A as well. You could use Visual Studio Configuration Manager dialog to declare the dependency between projects in the .sln file, but don't. Configuration Manager is a foreign concept in MSBuild, it's a legacy feature that shouldn't exist.

Instead, on your project reference in the .csproj file, set the ReferenceOutputAssembly metadata:

 Copy

```
<ProjectReference Include="..\ProjectA\ProjectA.csproj">
  <Project>{b402782f-de0a-41fa-b364-60612a786fb2}</Project>
  <Name>ProjectA</Name>
  <ReferenceOutputAssembly>>false</ReferenceOutputAssembly>
</ProjectReference>
```

This will indicate a dependency between projects to MSBuild, but won't pass the output assembly of ProjectA as a reference to the compiler when compiling ProjectB.

If you set ReferenceOutputAssembly to false you will notice that MSBuild stops copying the assembly A into the output directory for assembly B. If you would still like to copy assembly A to the output of assembly B (without referencing it), use this:

 Copy

```
<ProjectReference Include="..\ProjectA\ProjectA.csproj">
  <Project>{b402782f-de0a-41fa-b364-60612a786fb2}</Project>
  <Name>ProjectA</Name>
  <ReferenceOutputAssembly>>false</ReferenceOutputAssembly>
  <OutputItemType>Content</OutputItemType>
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
</ProjectReference>
```

This will produce an item of type Content in project B that includes the output assembly of project A, and the CopyToOutputDirectory metadata on that item will be set to Copy If Newer. This achieves copying a dependency without referencing it.

**Update:** many people have asked how to also copy the .pdb file in addition to the .exe/.dll. Turns out there is a trick, but it doesn't work when building in VS. But it's OK since you don't really need the .pdb in VS anyway, since the debugger will find the .pdb at its original path anyway.

Add this line to the project reference to also copy the .pdb:



```
<ProjectReference Include="..\ProjectA\ProjectA.csproj">
  <Project>{b402782f-de0a-41fa-b364-60612a786fb2}</Project>
  <Name>ProjectA</Name>
  <ReferenceOutputAssembly>>false</ReferenceOutputAssembly>
  <OutputItemType>Content</OutputItemType>
  <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
  <Targets>Build;DebugSymbolsProjectOutputGroup</Targets>
</ProjectReference>
```