

## How to use {smartassembly} with ClickOnce?

If you process your Assembly after your project is built, you will not be able to use ClickOnce to deploy it, because {smartassembly} has modified the assembly.

The solution is to integrate {smartassembly}'s process directly into the Build+Publish process.

1. First, you need to create your project with the GUI mode of {smartassembly}. The main assembly must be the assembly in the `\obj\Release` folder, and NOT the one in `\bin\Release`. The destination assembly will be in a temp folder, for example `'c:\temp\MyClickOnceApp.exe'`.
2. Then, you need to edit your C#/VB.NET project with an xml editor (or even notepad). The project file looks like this:

```
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    ...
  </PropertyGroup>
  ...
  <ItemGroup>
    ...
  </ItemGroup>
  <Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  <Target Name="BeforeBuild">
  </Target>
  <Target Name="AfterBuild">
  </Target>
  -->
</Project>
```

You need to add a few lines after the `<Import Project="...">` bloc:

```
<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <PropertyGroup>
    ...
  </PropertyGroup>
  ...
  <ItemGroup>
    ...
  </ItemGroup>
  <Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" />
  <!-- To modify your build process, add your task inside one of the targets below and uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  -->

  <PropertyGroup>
    <SmartAssemblyPath>"C:\Program Files\{smartassembly}\{smartassembly}.com"</SmartAssemblyPath>
  </PropertyGroup>

  <Target Name="BeforeBuild" Condition=" '$(Configuration)' == 'Release' ">
    <CreateProperty Value="true">
      <Output TaskParameter="Value" PropertyName="RunSmartAssembly"/>
    </CreateProperty>
  </Target>

  <Target Name="AfterCompile" Condition=" '$(RunSmartAssembly)' != '' ">
    <Exec Command="$(SmartAssemblyPath) [path]\MyProject.{sa}proj" />
    <Copy SourceFiles="C:\temp\MyClickOnceApp.exe" DestinationFiles="[path]\obj\Release\MyClickOnceApp.exe" />
  </Target>

</Project>
```

The first Xml node defines the path to {smartassembly}.com, the command line version of {smartassembly}. The console application is in the same folder than {smartassembly}.exe, usually in `C:\Program Files\{smartassembly}\{smartassembly}.com`

```
<PropertyGroup>
  <SmartAssemblyPath>"C:\Program Files\{smartassembly}\{smartassembly}.com"</SmartAssemblyPath>
</PropertyGroup>
```

The second Xml node tells MSBuild to set the *RunSmartAssembly* property in the *BeforeBuild* event, when the project is *fully* built in *Release* mode. This is necessary because Visual Studio is always compiling in the background:

```
<Target Name="BeforeBuild" Condition="'$(Configuration)' == 'Release' ">
  <CreateProperty Value="true">
    <Output TaskParameter="Value" PropertyName="RunSmartAssembly" />
  </CreateProperty>
</Target>
```

The last Xml node tells MSBuild to execute a few things in the *AfterCompile* event, if the *RunSmartAssembly* property is set:

```
<Target Name="AfterCompile" Condition="'$(RunSmartAssembly)' != "" ">
  <Exec Command="$(SmartAssemblyPath) SAPROJ_FILENAME" />
  <Copy SourceFiles="TEMP_FILENAME" DestinationFiles="OBJ_RELEASE_FILENAME" />
</Target>
```

The first thing to do is for {smartassembly} console to process the assembly (the one in the *obj\Release* folder). *SAPROJ\_FILENAME* must of course be replaced by the {sa}proj file name (with the full path).

Then, the assembly created by {smartassembly} is copy into the *\obj\Release* folder, to overwrite the assembly created by Visual Studio. *TEMP\_FILENAME* must be the full path to the assembly created by {smartassembly}, while *OBJ\_RELEASE\_FILENAME* is the path to the assembly created by Visual Studio and used as the main assembly in the {smartassembly} project.