Use ms build:

<?xml version="1.0" encoding="utf-8"?>

<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<Project Sdk="Microsoft.NET.Sdk">

<Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" />

## **Add a target and a task**

<Target Name="HelloWorld">

</Target>

<Target Name="HelloWorld">

<Message Text="Hello"></Message> <Message Text="World"></Message>

</Target>

## **Build the target**

<PropertyGroup>

...

<ProductVersion>10.0.11107</ProductVersion>

<SchemaVersion>2.0</SchemaVersion>

<ProjectGuid>{30E3C9D5-FD86-4691-A331-80EA5BA7E571}</ProjectGuid>

<OutputType>WinExe</OutputType>

...

</PropertyGroup>

<TargetFrameworkVersion>v4.5</TargetFrameworkVersion>

## **property value**

<Target Name="HelloWorld">

<Message Text="Configuration is $(Configuration)" />

<Message Text="MSBuildToolsPath is $(MSBuildToolsPath)" />

</Target>

### **Conditional properties**

<Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>

<Message Text="%24(Configuration) is %22$(Configuration)%22" />

## **Build items**

<ItemGroup>

<Compile Include="Program.cs" />

<Compile Include="Properties\AssemblyInfo.cs" />

</ItemGroup>

<ItemGroup>

<Compile Include="Program.cs;Properties\AssemblyInfo.cs" />

</ItemGroup>

### **metadata**

<Message Text="Compile Filename: %(Compile.Filename)" />

<Message Text="Backup files: @(Compile->'%(filename).bak')" />

## **BuildInParallel**

<PropertyGroup>

<BuildInParallel Condition="'$(BuildInParallel)' ==

''">true</BuildInParallel>

</PropertyGroup>

<MSBuild

Projects="@(\_MSBuildProjectReferenceExistent)"

Targets="GetTargetPath"

BuildInParallel="$(BuildInParallel)"

Properties="%(\_MSBuildProjectReferenceExistent.SetConfiguration);

%(\_MSBuildProjectReferenceExistent.SetPlatform)"

Condition="'@(NonVCProjectReference)'!='' and

('$(BuildingSolutionFile)' == 'true' or

'$(BuildingInsideVisualStudio)' == 'true' or

'$(BuildProjectReferences)' != 'true') and

'@(\_MSBuildProjectReferenceExistent)' != ''"

ContinueOnError="!$(BuildingProject)">

<Output TaskParameter="TargetOutputs"

ItemName="\_ResolvedProjectReferencePaths"/>

</MSBuild>

**Build Proect**

<ItemDefinitionGroup>

<PreBuildEvent>

<Command>copy $(ProjectDir)main.cpp $(ProjectDir)copyOfMain.cpp</Command>

<Message>Making a copy of main.cpp </Message>

</PreBuildEvent>

<PreLinkEvent>

<Command>copy $(ProjectDir)$(Configuration)\main.obj $(ProjectDir)$(Configuration)\copyOfMain.obj</Command>

<Message>Making a copy of main.obj</Message>

</PreLinkEvent>

<PostBuildEvent>

<Command>copy $(ProjectDir)$(Configuration)\$(TargetFileName) $(ProjectDir)$(Configuration)\copyOfMyproject.exe</Command>

<Message>Making a copy of myproject.exe</Message>

</PostBuildEvent>

</ItemDefinitionGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Release|Win32'">

<PreBuildEventUseInBuild>true</PreBuildEventUseInBuild>

<PreLinkEventUseInBuild>true</PreLinkEventUseInBuild>

<PostBuildEventUseInBuild>true</PostBuildEventUseInBuild>

</PropertyGroup>

<PropertyGroup Condition="'$(Configuration)|$(Platform)'=='Debug|Win32'">

<PreBuildEventUseInBuild>false</PreBuildEventUseInBuild>

<PreLinkEventUseInBuild>false</PreLinkEventUseInBuild>

<PostBuildEventUseInBuild>false</PostBuildEventUseInBuild>

</PropertyGroup>

**Property**

<PropertyGroup> <BuildDependsOn> BeforeBuild; CoreBuild; AfterBuild </BuildDependsOn> </PropertyGroup> <Target Name="Build" Condition=" '$(\_InvalidConfigurationWarning)' != 'true' " DependsOnTargets="$(BuildDependsOn)" Returns="@(TargetPathWithTargetPlatformMoniker)" />

<PropertyGroup>

<CoreBuildDependsOn>

BuildOnlySettings;

PrepareForBuild;

PreBuildEvent;

ResolveReferences;

PrepareResources;

ResolveKeySource;

Compile;

ExportWindowsMDFile;

UnmanagedUnregistration;

GenerateSerializationAssemblies;

CreateSatelliteAssemblies;

GenerateManifests;

GetTargetPath;

PrepareForRun;

UnmanagedRegistration;

IncrementalClean;

PostBuildEvent

</CoreBuildDependsOn>

</PropertyGroup>

<Target

Name="CoreBuild"

DependsOnTargets="$(CoreBuildDependsOn)">

<OnError ExecuteTargets="\_TimeStampAfterCompile;PostBuildEvent" Condition="'$(RunPostBuildEvent)'=='Always' or '$(RunPostBuildEvent)'=='OnOutputUpdated'"/>

<OnError ExecuteTargets="\_CleanRecordFileWrites"/>

</Target>

## **Targets and tasks**

<?xml version="1.0" encoding="utf-8"?>

<Project ToolsVersion="15.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<Project Sdk="Microsoft.NET.Sdk">

**Command**

msbuild SlnFolders.sln -t:NotInSolutionfolder:Rebuild;NewFolder\InSolutionFolder:Clean

**Code**

<Project DefaultTargets = "Compile"

xmlns="http://schemas.microsoft.com/developer/msbuild/2003" >

<!-- Set the application name as a property -->

<PropertyGroup>

<appname>HelloWorldCS</appname>

</PropertyGroup>

<!-- Specify the inputs by type and file name -->

<ItemGroup>

<CSFile Include = "consolehwcs1.cs"/>

</ItemGroup>

<Target Name = "Compile">

<!-- Run the Visual C# compilation using input files of type CSFile -->

<CSC

Sources = "@(CSFile)"

OutputAssembly = "$(appname).exe">

<!-- Set the OutputAssembly attribute of the CSC task

to the name of the executable file that is created -->

<Output

TaskParameter = "OutputAssembly"

ItemName = "EXEFile" />

</CSC>

<!-- Log the file name of the output file -->

<Message Text="The output file is @(EXEFile)"/>

</Target>

</Project>

<Project DefaultTargets = "Compile"

xmlns="http://schemas.microsoft.com/developer/msbuild/2003" >

<!-- Set the application name as a property -->

<PropertyGroup>

<appname>HelloWorldVB</appname>

</PropertyGroup>

<!-- Specify the inputs by type and file name -->

<ItemGroup>

<VBFile Include = "consolehwvb1.vb"/>

</ItemGroup>

<Target Name = "Compile">

<!-- Run the Visual Basic compilation using input files of type VBFile -->

<VBC

Sources = "@(VBFile)"

OutputAssembly= "$(appname).exe">

<!-- Set the OutputAssembly attribute of the VBC task

to the name of the executable file that is created -->

<Output

TaskParameter = "OutputAssembly"

ItemName = "EXEFile" />

</VBC>

<!-- Log the file name of the output file -->

<Message Text="The output file is @(EXEFile)"/>

</Target>

</Project>

<Target Name="Build" Inputs="@(Compile)" Outputs="$(OutputPath)$(AssemblyName).exe">

<MakeDir Directories="$(OutputPath)" Condition="!Exists('$(OutputPath)')" />

<Csc Sources="@(Compile)" OutputAssembly="$(OutputPath)$(AssemblyName).exe" />

</Target>

**Visual studio**

using System;

using System.Collections.Generic;

using System.Text;

namespace $nameSpace$

{

public class $className$

{

static void Main(string[] args)

{

Console.WriteLine("Hello VSX!!!");

Console.ReadKey();

}

}

}

<?xml version="1.0" encoding="utf-8" ?>

<Project DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

<PropertyGroup>

<Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>

<SchemaVersion>2.0</SchemaVersion>

<ProjectGuid></ProjectGuid>

<OutputType>Exe</OutputType>

<RootNamespace>MyRootNamespace</RootNamespace>

<AssemblyName>MyAssemblyName</AssemblyName>

<EnableUnmanagedDebugging>false</EnableUnmanagedDebugging>

</PropertyGroup>

<PropertyGroup Condition=" '$(Configuration)' == 'Debug' ">

<DebugSymbols>true</DebugSymbols>

<OutputPath>bin\Debug\</OutputPath>

</PropertyGroup>

<PropertyGroup Condition=" '$(Configuration)' == 'Release' ">

<DebugSymbols>false</DebugSymbols>

<OutputPath>bin\Release\</OutputPath>

</PropertyGroup>

<ItemGroup>

<Reference Include="mscorlib" />

<Reference Include="System" />

<Reference Include="System.Data" />

<Reference Include="System.Xml" />

</ItemGroup>

<ItemGroup>

<Compile Include="AssemblyInfo.cs">

<SubType>Code</SubType>

</Compile>

<Compile Include="Program.cs">

<SubType>Code</SubType>

</Compile>

</ItemGroup>

<Import Project="$(MSBuildBinPath)\Microsoft.CSharp.targets" />

</Project>

using System;

using System.Collections.Generic;

using Microsoft.VisualStudio.Project;

namespace SimpleProject

{

public class SimpleProjectNode : ProjectNode

{

private SimpleProjectPackage package;

public SimpleProjectNode(SimpleProjectPackage package)

{

this.package = package;

}

public override Guid ProjectGuid

{

get { return SimpleProjectPackage.guidSimpleProjectFactory; }

}

public override string ProjectType

{

get { return "SimpleProjectType"; }

}

public override void AddFileFromTemplate(

string source, string target)

{

this.FileTemplateProcessor.UntokenFile(source, target);

this.FileTemplateProcessor.Reset();

}

}

}

public override void AddFileFromTemplate(

string source, string target)

{

string nameSpace =

this.FileTemplateProcessor.GetFileNamespace(target, this);

string className = Path.GetFileNameWithoutExtension(target);

this.FileTemplateProcessor.AddReplace("$nameSpace$", nameSpace);

this.FileTemplateProcessor.AddReplace("$className$", className);

this.FileTemplateProcessor.UntokenFile(source, target);

this.FileTemplateProcessor.Reset();

}

**Properties**

**<?xml version="1.0" encoding="utf-8" ?>** <Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003"> <PropertyGroup> <MyReleaseOutput>.\release</MyReleaseOutput> </PropertyGroup> <Target Name="Build"> <Message Text="Building msbuildintro" /> <MSBuild Projects="msbuildintro.csproj" Targets="Build" /> </Target> <Target Name="Release" DependsOnTargets="Build"> <MakeDir Directories="$(MyReleaseOutput)" /> <Copy SourceFiles=".\bin\debug\msbuildintro.exe" DestinationFolder="$(MyReleaseOutput)" /> </Target> </Project>

## **Items**

**<?xml version="1.0" encoding="utf-8" ?>** <Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003"> <PropertyGroup> <MyReleaseOutput>.\release</MyReleaseOutput> </PropertyGroup> <ItemGroup> <MyReleaseFiles Include=".\bin\debug\\*.\*" /> </ItemGroup> <Target Name="Build"> <Message Text="Building msbuildintro" /> <MSBuild Projects="msbuildintro.csproj" Targets="Build" /> </Target> <Target Name="Release" DependsOnTargets="Build"> <MakeDir Directories="$(MyReleaseOutput)" /> <Copy SourceFiles="@(MyReleaseFiles)" DestinationFolder="$(MyReleaseOutput)" /> </Target> </Project>

## **Condition**

**<?xml version="1.0" encoding="utf-8" ?>** <Project xmlns="http://schemas.microsoft.com/developer/msbuild/2003"> <PropertyGroup> <MyReleaseOutput>.\release</MyReleaseOutput> <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration> </PropertyGroup> <Target Name="Build"> <Message Text="Building msbuildintro $(Configuration)" /> <MSBuild Projects="msbuildintro.csproj" Targets="Build" /> </Target> <Target Name="Release" DependsOnTargets="Build" Condition="$(Configuration) == 'Release'"> <MakeDir Directories="$(MyReleaseOutput)" /> <ItemGroup> <MyReleaseFiles Include=".\bin\$(Configuration)\\*.\*" Exclude=".\bin\$(Configuration)\\*vshost.exe" /> </ItemGroup> <Copy SourceFiles="@(MyReleaseFiles)" DestinationFolder="$(MyReleaseOutput)" /> </Target> </Project>