

# Microsoft C# Compiler Action

This action executes the Microsoft .NET Framework C# command line compiler.

The screenshot shows the 'Microsoft C# Compiler' dialog box with the 'Files & Direct...' tab selected. The dialog has several sections: 'Input files' with fields for 'Project directory' (C:\Projects\SimpleProject\SimpleProject\), 'File(s) to compile' (\*.cs), and 'Find source files in subdirectories' with 'Sub-directory' and 'Files' fields. The 'Output files' section includes 'Documentation comments to XML file', 'Output file' (C:\Projects\SimpleProject\Output\SimpleProject.exe), and 'Target type' (Library). The 'Framework Version' section has radio buttons for 'Use default Framework version (v4.0)' and 'Use 64-bit tools', and a dropdown for 'v4.0'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

**Microsoft C# Compiler**

General Runtime **Files & Direct...** Options Debugging Assemblies Resources

**Input files**

Project directory: C:\Projects\SimpleProject\SimpleProject\

File(s) to compile: \*.cs

**Find source files in subdirectories**

Sub-directory Optional - blank = All subdirectories

Files eg. \*.cs

**Output files**

Documentation comments to XML file:

Output file: C:\Projects\SimpleProject\Output\SimpleProject.exe

Target type: Library

**Framework Version**

☒ Use default Framework version (v4.0) ☐ Use 64-bit tools

☐ Use v4.0

OK Cancel Help

On this page:

## Files & Directories

### Input Files

#### Project Directory

The directory in which the project to be compiled resides.

#### Files to compile

The list of files to compile.

### Find source files in sub-directories

#### Sub-directories

The sub-directories to search for referenced source files required for the compile.

#### Files

The file mask to use when looking in the sub-directories for source files. Typically left as \*.cs.

## Output files

### Documentation comments to XML file

Generate documentation based on XML documentation tags in the source code.

### Output file

The file that the resulting build should be written to.

### Target type

The form in which the output file should take. This allows specifying if the resulting file is an executable, library, etc.

## Framework Version

### Use default framework version (vX.X) / Use vX.X

Allows the selection of the .NET version to use for the tool. The minimum is .NET v2.0.

### Use 64-bit tools

This options specifies as whether to force the usage of 64-bit tools on a 64-bit system or not. As FinalBuilder is currently a 32-bit application it will default to running the 32-bit version on a 64-bit system. Check this option if you require the 64-bit version to be run.

## Options

## Optimization

### File align bytes

Specifies the size on which elements are aligned in the output file. Valid values are 512, 1024, 2048, 4096, and 8192. These values are in bytes.

### Optimize

Specifies that the common language runtime should optimise code at runtime.

## Pre-processor

### Define pre-processor symbols

Specify the symbols which should appear as pre-processor symbols in the compile.

## Miscellaneous

### Base address

The base address to load the the binary from.

### Code page

The language code page to use when reading all source code files.

### Main method

The name of the type that contains the main method.

### Enable incremental compilation

Performs an incremental compilation. (obsolete in newer compiler versions).

### Do not compile with the csc.rsp file

Instructs the compiler to not load the configuration file "csc.rsp" for this compilation.

## **Suppress compiler banner information**

Turns off the banner displaying for the command line compiler. This simply reduces the amount of information logged for the build.

## **Enable unsafe mode**

Allows the use of the unsafe keyword in the source which is being compiled.

## **Display compiler output using UTF-8 encoding**

The output from the compiler is generated with UTF-8 encoding.

# **Debugging**

## **Debugging/Error checking options**

### **Promote warnings to errors**

When a warning is raised it will be treated as an error. Unless specifically ignored warnings will cause the build to fail.

### **Check integer arithmetic at run time**

Check integer math to make sure results are not outside the bounds of the type.

### **Specify the absolute path to the file in compiler output**

When files are listed in the compiler output, this option will specify whether the full path should be used or not. Helpful in tracking down relative path issues.

### **Debug information**

The resulting files generated from the compile will contain debug information.

### **Set warning level**

The level of warnings to report from the compilation. The higher the value the more warnings that will be shown.

### **Suppress warnings**

Turn off seeing any warnings from the compilation process.

# **Assemblies**

## **Assemblies**

### **Do not import standard libraries**

The compiler will not import the standard libraries for the compile.

### **Modules to be part of this assembly**

A list of modules that are to be part of the assembly. Each module appears on a separate line within the dialogue.

### **Import metadata from a file that contains an assembly**

Tells the compiler to import public type information from the supplied assembly. This allows referencing metadata from the imported types as though they were part of the assembly.

### **Location of assemblies referenced by means of /reference**

The locations of the assemblies which are to be referenced.

# **Resources**

## **Resources**

### **Create link to a managed resources**

Generates a link to a managed resource.

### **Embed .NET framework resources into the output file**

The .NET resource for the project is embedded into the generated output file.

### **Insert .ico files into the output file**

The supplied icon file is embedded into the output file. Typically an executable.

### **Insert Win32 resources into the output file**

The Win32 resource for the project is embedded into the generated output file.