

# Using the .NET Wrapper

This chapter covers the following topics:

- Generation Process
  - Using .NET Wrapper Interactively
- 

## Generation Process

To generate the C# client or server code, use the *EntireX Workbench*. This can be done interactively with the graphical user interface or in *IDL Compiler Command-line Mode*.

## Using .NET Wrapper Interactively

To use the .NET Wrapper functions, open your Eclipse Workspace.

## Setting Wrapper Options

Before you start the generation of C# code for the first time, adjust the global options for the .NET Wrapper in the Eclipse preferences under **Software AG > EntireX > .NET Wrapper**.

On the **General** tab, set the paths to the Microsoft .NET Framework directory and the EntireX .NET Wrapper runtime (*SoftwareAG.EntireX.NETWrapper.Runtime.dll*). The preferences on the **Generate Client** and **Generate Server** tabs are identical. Choose your default settings for the client/server generation.

Option	Description
C# compiler options	Used to define additional options for the C# compiler (csc.exe).
Project relative output directory	A folder (structure) for C# code generation and compilation relative to the Eclipse project where the IDL file is located.
String handling	Default/String/StringBuilder: in the default case, "string" is used for IN and "StringBuilder" is used for OUT/INOUT parameters. In the case of "String", the C# type "string" is used for IN/INOUT/OUT. In the case of "StringBuilder", the C# class "StringBuilder" is used.
Class name prefix for inner classes	A string is used to prefix the name of the inner classes when the Sanitize option is selected (only necessary for Visual Basic clients).
Use IDL file base name for output	Use this flag only if you have large environments built with previous versions of the .NET Wrapper. If this flag is set and you have more than one library in your IDL file, a C# file is generated with the file base name of the IDL file (base name=file name without extension). If this flag is not set, the library name is used as file base name for the generated C# file (one file for every library in the IDL file).
Sanitize	If this flag is set, the IDL names are sanitized according to the programming conventions for C#. See <i>Mapping IDL Data Types to .NET Data Types</i> .
Generate "char" for A1 instead of String	The C# data type "char" is used for IDL parameters of type A1.
Generate "byte" for B1 instead of byte	The C# data type "byte" is used for IDL parameters of type B1.
Remove trailing blanks	Remove trailing blanks after unmarshalling the data. This flag is useful on the client side to remove trailing blanks before the data returned from the server is put into the C# classes <code>string</code>   <code>StringBuilder</code> .

These options are then used as default for the properties of your individual IDL files. You can change these options (except those on the **General** tab) for every individual IDL file.