

SBS-CASE Tools TIPDriverAPI

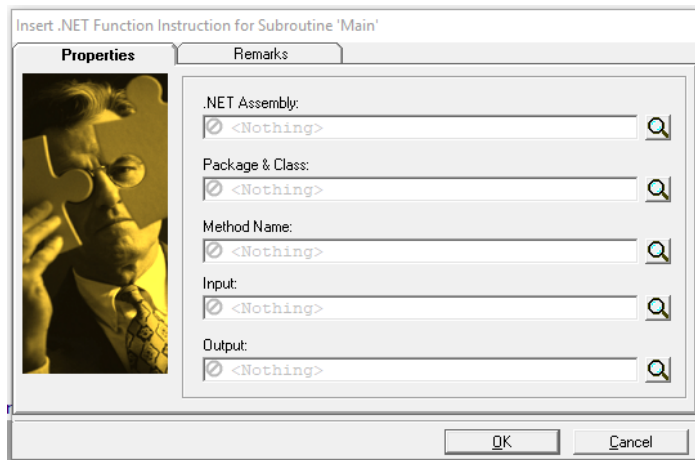
While SBS-CASE Tools contains many features that allow for rapid development of custom solutions for businesses, occasionally there arises a need to implement features that are not contained in the standard SBS-CASE Tools library. This problem can be addressed by using .NET code contained in a separate class library (.dll) through the “Invoke .NET method” instruction. This document is meant to describe the API through which methods in external class libraries can be invoked and provide instructions to create and deploy a class library using Visual Studio 2019.

API Description

TipDriverAPI (Static)
Static Private Member Variables
Static Public Methods (string, StringBuilder) Static Private Methods

The TipDriverAPI class is statically defined and is implemented through the “Invoke .NET method” instruction in SBS-CASE Tools. It can be accessed through static public methods with a void return type. Public methods should include two parameters; a string, which contains any input parameters, and a StringBuilder, which will contain any output.

Creating Invokable Class Libraries in Visual Studio 2019



The general format for creating a class library is included in the “TIPDriverAPI” solution. Class libraries created should target the **.NET Framework version 3.5**. The class that will be used to communicate with the TIP Driver should be statically defined.

The TIP Driver can only communicate with the static class through public methods with two parameters, a string and a StringBuilder. The string parameter will be the value passed in the

“Input” field in SBS-CASE Tools. The StringBuilder parameter will be the value returned to the TIP Driver and stored in the variable or structure provided in the “Output” field. Even if no input will be provided or output will be expected, public methods should include the two parameters.

The “Invoke .NET method” instruction requires five fields to be filled. The first requires a path to the .NET Assembly (.dll) that contains the API class. The second field requires the namespace and class name in the format “**namespace.class**”. The third field requires the name of the method to be invoked and does not require any arguments to be provided (as those will be provided in the next two fields). For a method defined in C# as “**public static void MyMethod (string input, StringBuilder output)**”, one would provide “**MyMethod**” in this field. The last two fields, input and output, will be the two arguments passed to the method, the first as the string parameter and the second as the StringBuilder parameter.