

QualiSystems TestShell™



TestShell Library Development

Software Development Kit

TestShell 4.7 and above

Document Release Date: October, 2012

Updated: March, 2014



Table of Contents

General.....3
Purpose3
Steps4

General

Purpose

The purpose of this document is to provide the basic guidelines on how to write a TestShell Library (version 4.7 and above) from MS Visual Studio 2010 and above.

Steps

1. Create a .NET DLL solution.
2. Add LibraryAttributes.cs (can be found under: "C:\Program Files (x86)\QualiSystems\TestShell\DriverBuilder\LibraryAttributes.cs") to your project. It holds all the TestShell specific attributes.
3. Use a public class to define your list of methods. The class name is the name of the session type in TestShell Studio, Do not use an existing TestShell session type (Dll, SSH, GUI etc...).
4. Optional: Inherit the class from ICancelable if required. This interface should be implemented to gracefully end session operations (requires additional method to be implemented under ICancelable).
5. In the AssmeblyInfo.cs:
 - Verify `[assembly: AssemblyVersion("1.0.0.1")]` has the proper version
 - Add this attribute `[assembly: Library(typeof(ClassName), "StudioStencil")]`. An example for a stencil we are using is "Traffic".
6. Method guidelines
 - Use primitive variables.
 - You can use ENUM variables (numeric and string only).
 - For a string vector use `string[]`, for a numeric vector use `double[]`.
 - For a string matrix use `string[,]`, for a numeric matrix use `double[,]`.

- Use `out` parameters if you have several parameters that you would like the method to return.
- Throw an exception of type `ErrorException` providing the following information:
 - Error name.
 - Description
 - Originating step (can be empty).
 - Parameters (will be represented as a string matrix in TestShell Studio for the user to extract required information).
- Method attributes:
 - `[Folder("Execution")]`
Specify the folder that will be displayed in TestShell Studio. The folder can be a hierarchy (e.g. `"Folder1\Folder2\Folder3"`).
 - `[Description("write the method description here")]`
Optional. Provide the method description that will be displayed in TestShell Studio.
 - `[Alias("Method Alias")]`
Optional. Provide the method alias that will be displayed in TestShell Studio. This string can have spaces in it as opposed to the .net method name.
 - `[StartSession]`
Optional. Add the attribute to specify a method to be called when a Studio session is created. All the method parameters will appear in the session section. Common usage of for this method is for connection functions.

- `[EndSession]`
Optional. Add the attribute to specify a method to be called when a Studio session ends.
- `[Hidden]`
Optional. Add this attribute when you wouldn't want the method to be exposed to the user.
- `[Cancelable]`
Optional. Specify the method is cancelable. Add this attribute if you've implemented the `ICancelable` interface.
- Parameter attributes:
 - General usage – add the below before the parameter
`[Description("The parameter description"), Mandatory, Alias("A parameter description"), ParameterDefaultValue("True")]`
 - `Description`
Optional. Provide the variable description.
 - `Mandatory`
Optional. Add this attribute if the parameter is mandatory.
 - `Alias`
Optional. Provide the variable alias. This alias can contain spaces.
 - `ParameterDefaultValue`
Optional. When required add a default value.