Calling FIA from C# and Visual Studio

Andrew Colin

24th April 2017

Introduction

The purpose of this document is to provide, in detail, the steps required to set up a demonstration program that calls the Flametree FIA library from a C-Sharp program within the Windows operating system.

This document assumes that you have a licenced or evaluation copy of FIA.

FIA and SWIG

FIA is written in C++ for reasons of efficiency and portability. However, several of Flametree's customers prefer to work in C#. We have therefore provided a set of interface wrapper files, generated by the SWIG interface generation tool, that allows FIA's API to be called from C# code. More information on SWIG is available at *www.swig.org*.

Calling FIA from C-Sharp

This section describes how to set up a C-Sharp project from scratch. If you would prefer to use a C-Sharp project file that has already been set up, please refer to the next section.

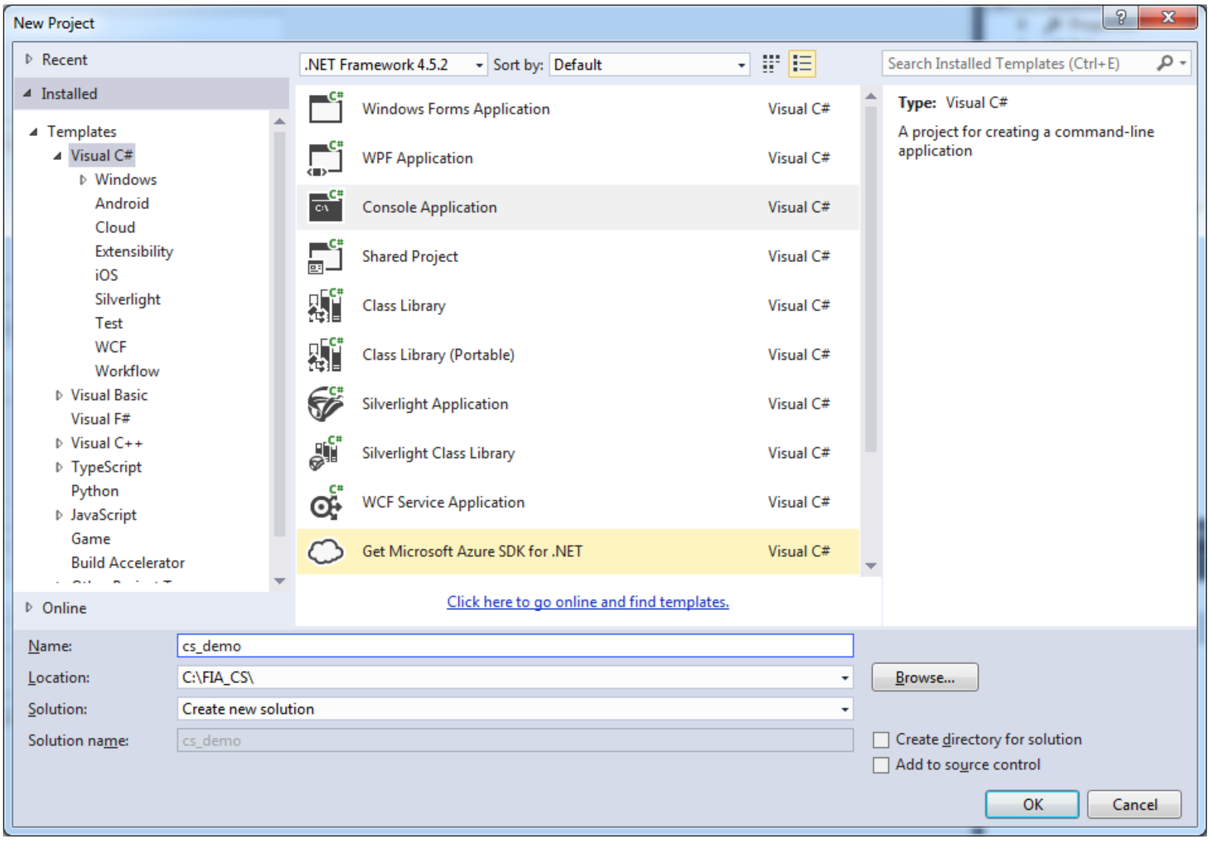
1. Create a directory for your project. Here, we have created a projedirectory called C:\FIA_CS.
2. Copy the required demo files to this directory:

demo_csharp.cs contains sample code showing how to call the FIA API from C#
enum.cs contains the values of various constants used by the FIA API
StringMatrix.cs, StringVector.cs, wrapper.cs and **wrapperPINVOKE.cs** are machine-generated files that are set up by the SWIG package. Please do not modify the contents of these files.
wrapper.dll is an interface file required to call FIA.

3. Copy the FIA executable and support libraries to your project directory, or ensure they are available on your system's path:

libfia64.dll is the core FIA library. Please do not distribute this file!
libxl.dll is FIA's Excel support file
In addition, all **pricing libraries** should be accessible by libfia64.dll. These libraries typically have prefix 'FT_' and names such as FT_cash.dll, FT_bill.dll, depending on the types of securities priced. If necessary, copy all such libraries to the project directory.

4. Open the Visual Studio IDE, click on File/New/Project. Under Visual C#/Windows, select 'Console Application', set the 'Location' (at the bottom of the window) to your new directory, enter a name (eg 'cs_demo') in the Name field, and uncheck 'Create directory for solution. Then press OK (see Figure 1).


New Project
Recent
Installed
Templates
Visual C#
Windows
Android
Cloud
Extensibility
iOS
Silverlight
Test
WCF
Workflow
Visual Basic
Visual F#
Visual C++
TypeScript
Python
JavaScript
Game
Build Accelerator
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
Windows Forms Application Visual C#
WPF Application Visual C#
Console Application Visual C#
Shared Project Visual C#
Class Library Visual C#
Class Library (Portable) Visual C#
Silverlight Application Visual C#
Silverlight Class Library Visual C#
WCF Service Application Visual C#
Get Microsoft Azure SDK for .NET Visual C#
Click here to go online and find templates.
Browse...
OK
Cancel

Type: Visual C#

A project for creating a command-line application

Name: cs_demo

Location: C:\FIA_CS\

Solution: Create new solution

Solution name: cs_demo

☐ Create directory for solution

☐ Add to source control

Figure 1: Creating a new project

At the upper-right hand of the IDE, you will now see the ‘Solution Explorer’.

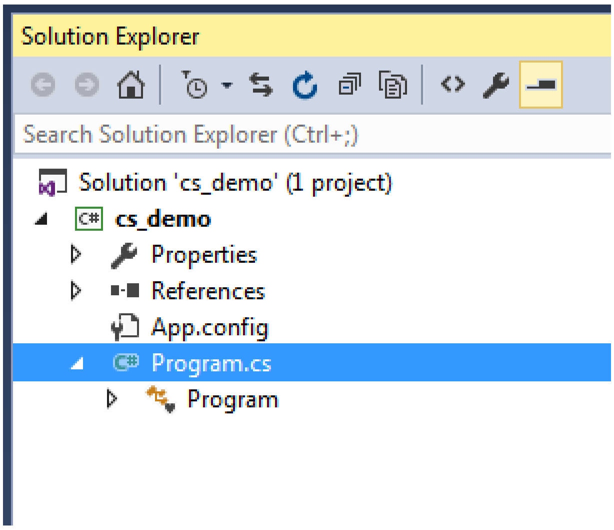
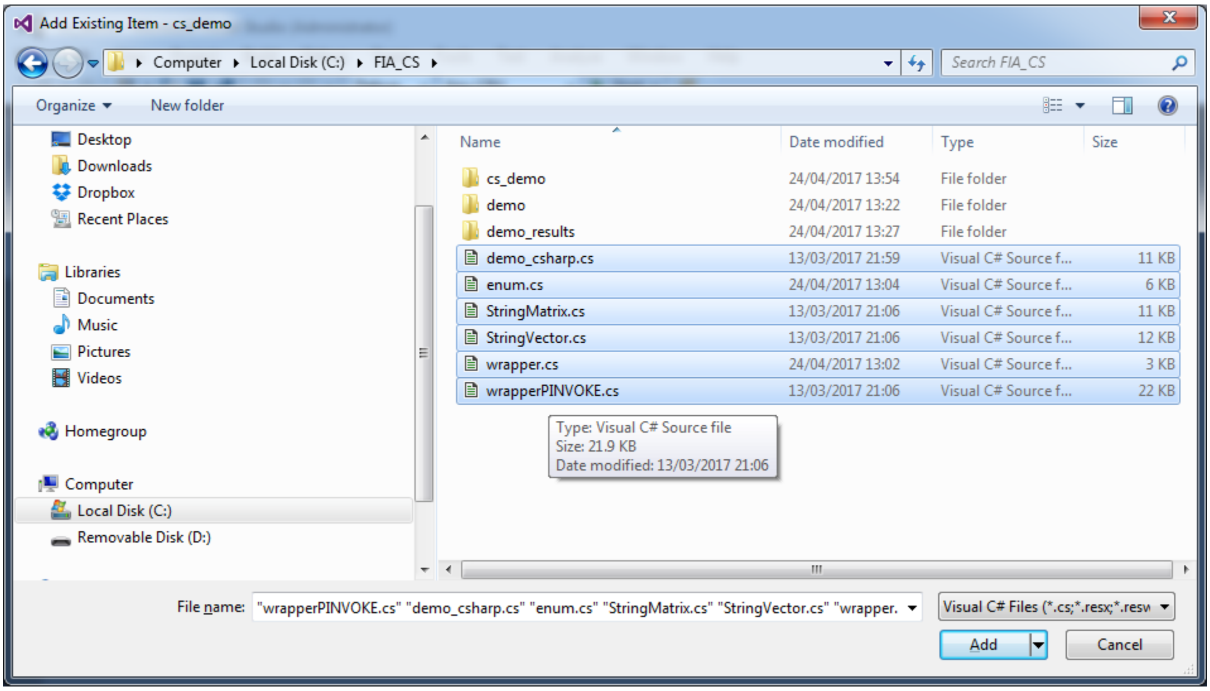


Figure 2: Solution explorer

Left-click on ‘Program.cs’ and choose ‘Delete’. This will erase the automatically generated code for this project, which we do not need.

Next, right-click on ‘cs_demo’ in the Solution Explorer and select ‘Add/Existing Item’. Navigate to your project directory and select all the cs files:



Name	Date modified	Type	Size
cs_demo	24/04/2017 13:54	File folder	
demo	24/04/2017 13:22	File folder	
demo_results	24/04/2017 13:27	File folder	
demo_csharp.cs	13/03/2017 21:59	Visual C# Source f...	11 KB
enum.cs	24/04/2017 13:04	Visual C# Source f...	6 KB
StringMatrix.cs	13/03/2017 21:06	Visual C# Source f...	11 KB
StringVector.cs	13/03/2017 21:06	Visual C# Source f...	12 KB
wrapper.cs	24/04/2017 13:02	Visual C# Source f...	3 KB
wrapperPINVOKE.cs	13/03/2017 21:06	Visual C# Source f...	22 KB

Type: Visual C# Source file
Size: 21.9 KB
Date modified: 13/03/2017 21:06

File name: "wrapperPINVOKE.cs" "demo_csharp.cs" "enum.cs" "StringMatrix.cs" "StringVector.cs" "wrapper.

Visual C# Files (*.cs;*.resx;*.resw

Figure 3: Selecting project files from project directory

Next, click on the down-arrow to the right of the ‘Add’ button and select ‘Add as link’:

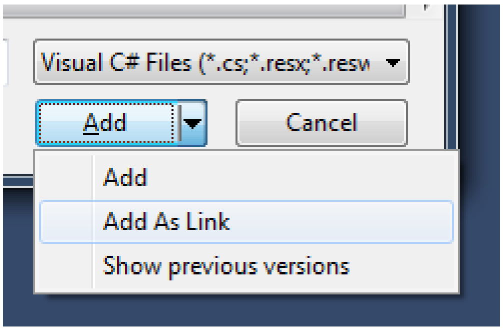


Figure 4: Adding links to files in project directory

Next, right-click on ‘cs_demo’ in the Solution Explorer and select ‘Properties’ at the bottom of the menu:

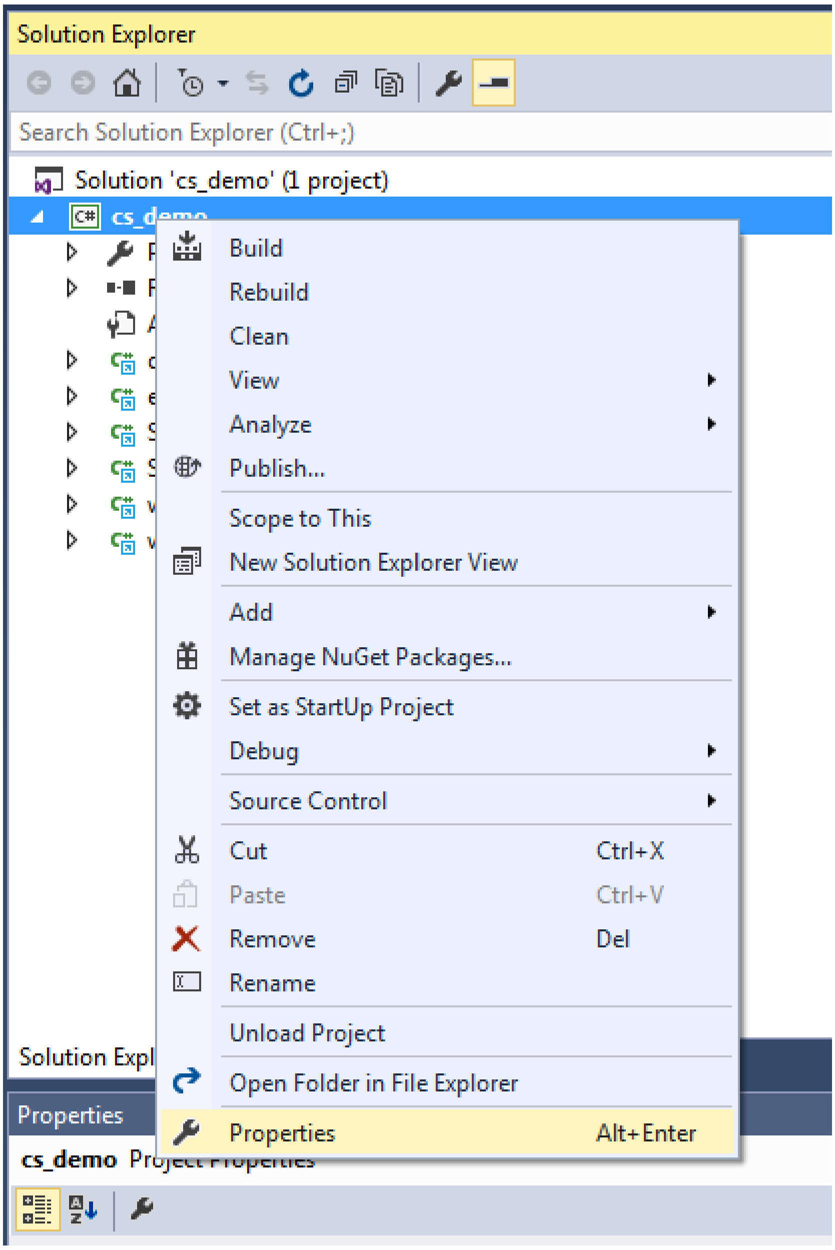
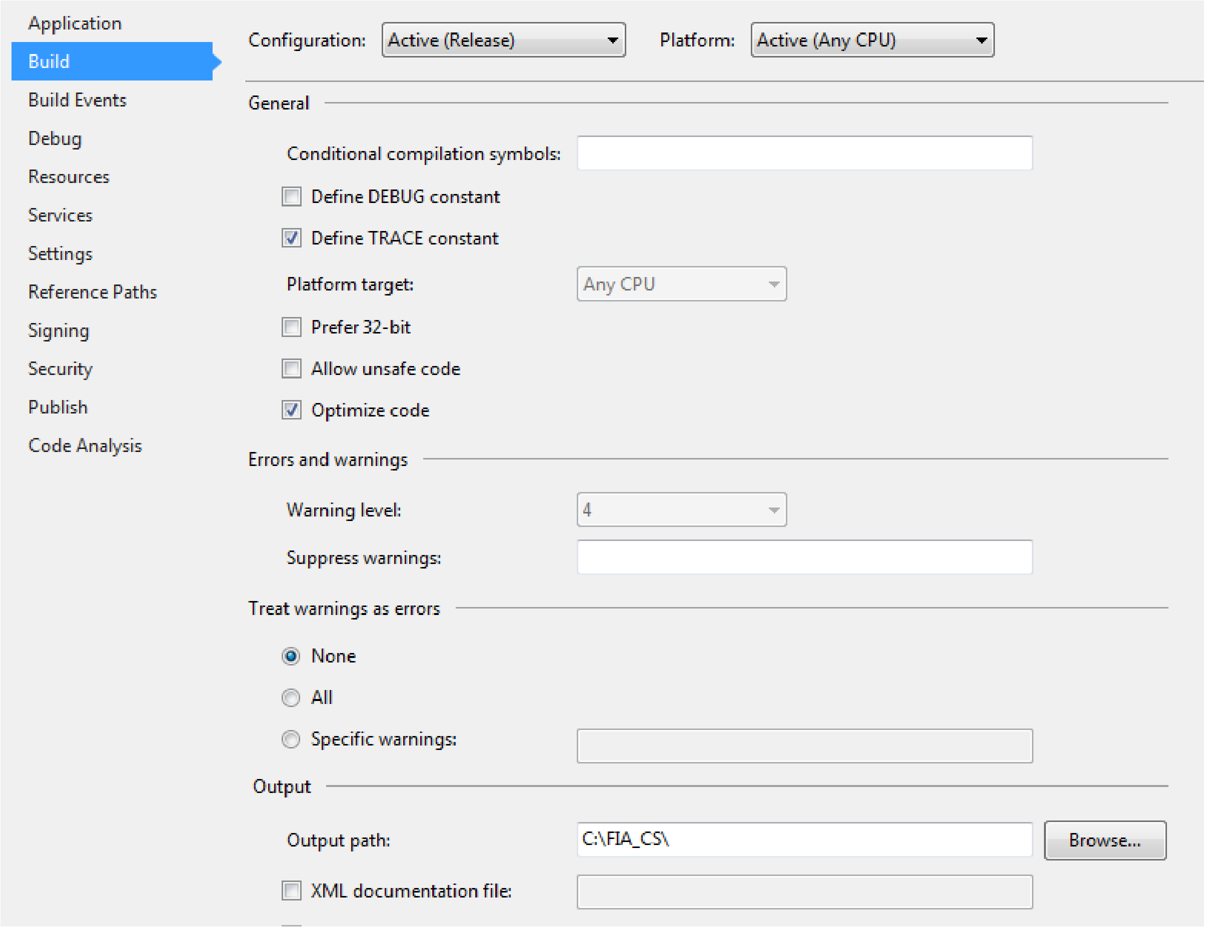


Figure 5: Setting project properties

In the ‘Build’ tab, do the following:

- Uncheck the ‘Prefer 32-bit code’ flag
- Change ‘Output path’ to your project directory



Configuration: Active (Release)
Platform: Active (Any CPU)

General

Conditional compilation symbols:

□ Define DEBUG constant

☑ Define TRACE constant

Platform target: Any CPU

□ Prefer 32-bit

□ Allow unsafe code

☑ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings:

Treat warnings as errors

● None

○ All

○ Specific warnings:

Output

Output path: C:\FIA_CS\

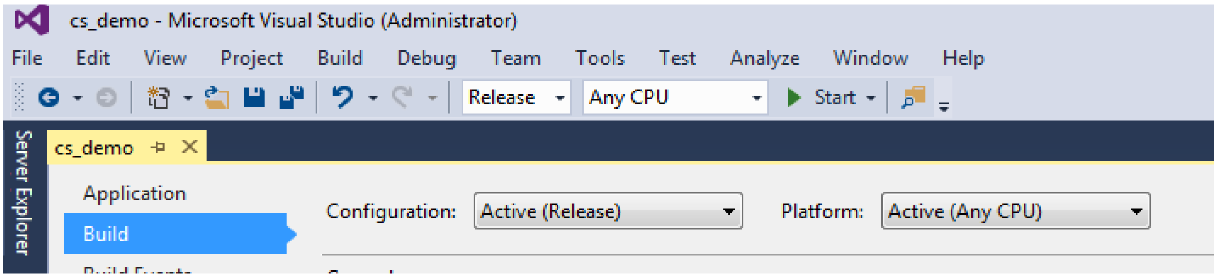
□ XML documentation file:

Figure 6: Setting the project configuration

Save your project by clicking on the floppy disk icon at the top left of the IDE window.

You can now compile and run your application from within the IDE.

To compile a release version, select ‘Release’ from the select box at the top of the window, and click on ‘Build/Build Solution’.

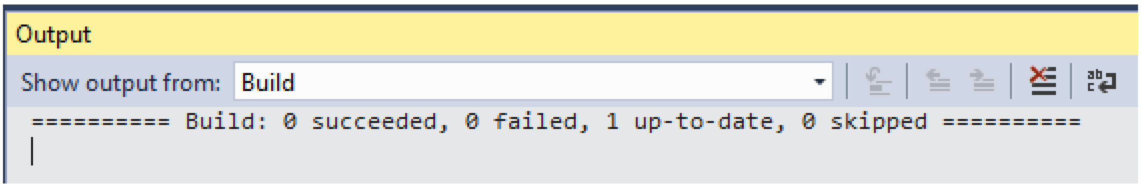


Release
Any CPU

Configuration: Active (Release)
Platform: Active (Any CPU)

Figure 7: Setting the build configuration

If the project has compiled successfully, the panel at the lower left of the IDE will look as follows:



```
========== Build: 0 succeeded, 0 failed, 1 up-to-date, 0 skipped ==========
```

Figure 8: Result of a successful build

You can now run the project, either by clicking on the green 'Start' button at the top of the IDE window, or by opening a command console (or DOS prompt) and typing 'cs_demo'

Pre-defined project files

A predefined VS project file is included with the other demo files. All files are available at section 'Downloads' at our website at www.flametreetechnologies.com.