

Getting started using the included software

Daniel J. Duffy and Andrea Germani
C# For Financial Markets, Wiley

April 2014

IMPORTANT DISCLAIMER. *The code is for demonstration purposes only, it comes 'AS IS' with NO WARRANTY AND GUARANTEE. No liability is accepted by the authors for the use of this code in any circumstances with respect, including, but not limited to any kind of direct, indirect, incidental damaged, loss of data, lost profits, business interruption, etc. Any market parameters used are not real data but have been created to clarify the exercises and should not be viewed as actual market data. If you don't accept and agree with the disclaimer do not download and do not use the software.*

The software has been developed using: Windows XP and 7; Visual Studio 2005, 2008, 2010; .Net Framework 4.0; Excel 2003 and 2007.

1 Introduction

The archive **CsForFinancialMarkets.zip** contains two main directories with two different stand alone approaches to use the same example code of the book.

- **CsForFinancialMarkets**: it contains one big solution with many projects. Examples are sorted by the chapter number. Reusable parts of code are organized in small libraries (**.dll**) which are referenced in examples. To make work examples, you first need to build the **.dll**. Section 2 explains in details how to make this framework work.
- **CsForFinancialMarketsLight**: it contains a directory for each chapter. Each directory contains a solution with stand alone projects. The **.cs** files used in each project are not a copy of the files but a link to the code in **CsForFinancialMarkets** (this ensure to use the same code). For this reason do not change the relative position of the two main directories. Also reusable parts of code are linked to the project, so no preliminary **.dll** building is need. For a quick start this second framework is easier and better. It allows the quick use of the example without building all the projects and the navigation thought the code used in the example. See the example in Section 3.

Preliminary step for both approaches: download additional libraries from the Web and carefully read Licensing notice.

- Download ALGLIB from www.alglib.net and read carefully Licensing notice. Choose the C# version 3.6.0 and save **Alglibnet2.dll** under **CsForFinancialMarkets\Lib**.
- Download Excel-Dna Version 0.30 from <http://exceldna.codeplex.com>. You need the following two files:
 - **ExcelDna.Integration.dll**
 - **ExcelDna.xll**

Put them under **CsForFinancialMarkets\Lib**.

- Download LinqToExcel Version 1.7.0 library from <http://code.google.com/p/linqtoexcel> and read carefully Licensing notice. Copy the **.dlls** in **CsForFinancialMarkets\Lib**.

2 Using the material in CsForFinancialMarkets directory

1. Unzip the `CsForFinancialMarkets.zip` archive file. You can extract files anywhere, but in this tutorial we are going to put them directly under `C:\`. Open the directory `CsForFinancialMarkets`: its content is shown in Figure 1.

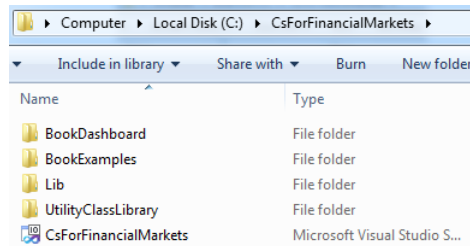


Figure 1: First level directories and files

The main directories are:

- **BookDashBoard**: a Console Project running a menu to choose between all book examples
 - **BookExamples**: containing each single book example as ready to run Console Project
 - **Lib**: container for downloaded third parties libraries and tools (.dll and .xll)
 - **UtilityClassLibrary**: some reusable libraries from the book
2. Open `CsForFinancialMarkets` solution and navigate to solution folders.
 - Opening the main solution. Double-click on the file `CsForFinancialMarkets.sln`, located in the main directory as shown in Figure 1. The entire solution will be opened (Visual Studio must be installed). If you are using Windows 7, you need to run Visual Studio as administrator.
 - Navigating to solution folders. From the **Solution Explorer** window, you can see all the projects contained in the solution: they are organized in Solution Folders. Figure 2 show the first directory level. If you click on each Solution Folder (**BookExamples** and **UtilityClassLibrary**) you can find more sub-folders containing projects.

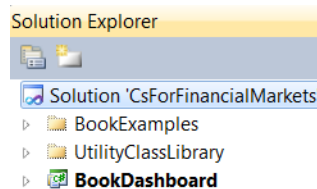


Figure 2: Solution Folders

3. Check that the references of all projects are working. How to check it? For each project, go to **References** in the **Solution Explorer** and verify that no yellow exclamation mark appears, as in Figure 3.

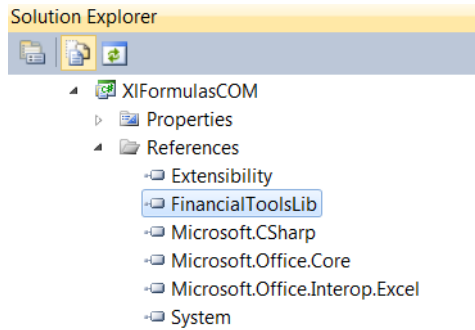


Figure 3: Example of correct references

If yellow exclamation mark appears it means that there is a broken reference, namely the file path set does not contain the file referenced, as in Figure 4.

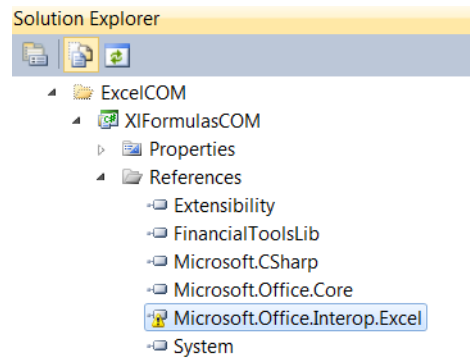


Figure 4: Example of broken reference (Microsoft.Office.Interop.Excel)

In this case you should check the reference location. Let's show how to do it. Consider as an example the Microsoft.Office.Interop.Excel reference.

- Select the reference Microsoft.Office.Interop.Excel, right-click and select Remove.
- Select References under the needed project in the Solution Explorer, right-click and select Add Reference....
- Click on COM tab as shown in Figure 5 and look for Microsoft Excel 12.0 Object Library.

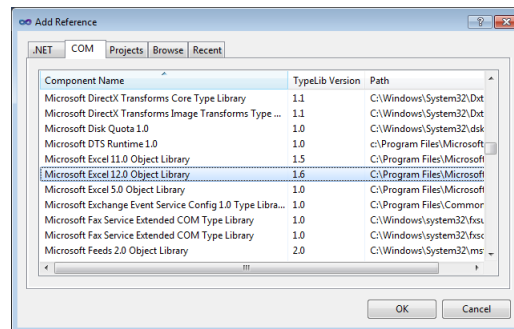


Figure 5: Adding the reference to Microsoft.Office.Interop.Excel

4. Select the solution `CsFinancialMarkets` from **Solution Explorer**, then right-click and select **Build** or **F6**.
5. How to use `CsFinancialMarkets`. There are several ways to use the material included in the solution.

- *Using the single menu to navigate through the book's examples.* Select the project `BookDashboard`, check that it is the **StartUp Project**: the name of the project should be in bold as in Figure 2. Otherwise right-click and select **Set as StartUp Project**. Type **Ctrl+F5** to run the example list.

A quicker way is to double-click directly on `CsForFinancialMarkets\BookDashboard\bin\Debug\BookDashboard.exe`

- *Running a single console example project.* Alternatively a single example project can be run directly: from **Solution Explorer**, click the solution folder named `BookExamples`, choose the chapter you are looking for and select the project you need. Right-click on it and select **Set as StartUp Project**. Finally type **Ctrl+F5**.
- *Examine the code of reusable libraries.* The solution folder `UtilityClassLibrary` contains several reusable class libraries presented in the book. They are used as references for example projects contained in the solution folder `BookExamples`: using the debugging it is possible to analyse the used source code from a selected example.

As a demonstration, let's show how to examine the class `STFut` used in the `Example1` of Chapter 14.

- Double-click on `BookExample`, select `Ch14 Short Term Interest Rate (STIR) Futures and Options`, select the project and set it as start up (right-click **Set as StartUp Project**).
- In the `Example1()` code, select class `STFut` and press **F12** to go to the source code of the selected class, see Figure 6a.
- Add some **Breakpoints** in the desired positions, as in Figure 6b. Finally, you can start debugging using **F5** and using **Step Into (F10)**, **Step Over (F11)**, etc.

```
// Class STFut: Property and Method
public static void Example1()
{
    // my Future
    STFut myFut = new STFut(99.00, 12, 2013);
    class STFut
    Console.WriteLine("fwdDF(360): {0}", myFut.fwdFq(360));
    Console.WriteLine("IMMDate: {0}", myFut.GetIMMDate);
}
```

(a) Selecting the class

```
329 public class STFut: IPrice
330 {
331     // data member
332     private double mktPrice;
333     private double implRate;
334     private Date IMMDate;
335     private Date NotionalRepayDate;
336
337     // Constructor
338     public STFut(double price, int Month, int Year)
339     {
340         this.mktPrice = price;
341     }
}
```

(b) Using Breaking Points

Figure 6: Example: examining the code with the debugger

- *Using Excel Spreadsheets.* Under the solution folder `BookExamples` you can find some Excel spreadsheets referring Chapters 12, 22 and Appendix 4. To use them correctly, you need to complete also step 7 and 8 below. You can use the spreadsheets: calling them from `BookDashboard` project, or from the project containing them under `BookExamples` (running the single console project or simply clicking on the spreadsheet).
6. When you use **Managed Automation Add-In** in Excel spreadsheet (Chapters 12 and 22), you need to add your class to the list of installed automation add-ins. Moreover, if you use the class from Excel VBA, you need to check the reference to your class. Here a quick guide for the specific files.

Using Excel 2003

- *BondBasic.xls.*
Tools->Add-Ins->Automation, select `XLFormulas.XLBond` from the list.

- *BondLoadInMemory.xls* and *BondSerialization.xls*.

Tools->Add-Ins->Automation, select *XLFormulas.XLBond* from the list. Open VBA from Excel (Alt+F11), go to Tools->References and select *XLFormulas*. Browse and select the file location from: C:\CsForFinancialMarkets\UtilityClassLibrary\Automation Add-In XLFormulas\bin\Debug\XLFormulas.tlb.

- *MultiCurveCOM.xls*.

Tools->Add-Ins->Automation, select *XLRateCurve.XLRate* from the list. Open VBA from Excel (Alt+F11), go to Tools->References and select *XLRateCurveCOM*. The file location is: C:\CsForFinancialMarkets\UtilityClassLibrary\XLRateCurveCOM\bin\Debug\XLRateCurveCOM.tlb

Using Excel 2007

- *BondBasic.xls*, *BondLoadInMemory.xls* and *BondSerialization.xls*.

Microsoft Office Button->Excel Options->Add-Ins. In the Managed list select Excel Add-ins and click Go. In the Add-Ins dialog box, click Automation, select the class *XLFormulas.XLBond* then click Ok. Finally click Ok.

- *MultiCurveCOM.xls*.

As for previous files, but here select the class *XLRateCurve.XLRate*.

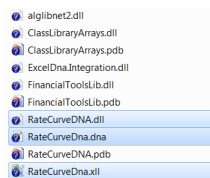
7. When you use *RateCurveDNA.xls* and *CapFloorSwaption.xls* (Chapter 22 and Appendix 4), you need to register the Excel-DNA .xll. The registration is done automatically with the VBA code in the *Workbook_Open()* of each spreadsheet. Open VBA from Excel (Alt+F11) to see the code, as in Figure 7. If you move the spreadsheet you need to check and update the .xll registration path.

```
Private Sub Workbook_Open()
    Application.ScreenUpdating = False

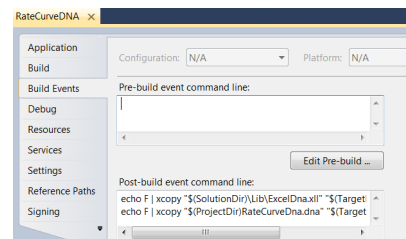
    Dim wbPath As String
    wbPath = ActiveWorkbook.Path
    ' If you move the file you should change this path
    ' Load .xll from correct path.
    ok = Application.RegisterXLL(wbPath + "\bin\Debug\RateCurveDna.xll")
    If (ok = False) Then
        MsgBox "xll NOT loaded, please check the path."
    End If
End Sub
```

Figure 7: Example of *Workbook.Open()*

Moreover, note that the .dna, .xll and .dll files should be in the same directory, as in the Figure 8a. The project *RateCurveDNA* will copy (in post-build event) relevant files in the same directory (check it: in the project select Properties, Build Events and then Post-build event command line), see Figure 8b. Note that *RateCurveDNA.xll* is a renamed copy of *ExcelDna.xll* (see details in the general Excel-DNA documentation <http://exceldna.codeplex.com>). Alternatively, files can be moved also manually in the desired position.



(a) Selected files in the same directory

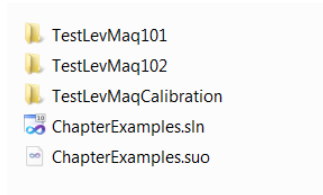


(b) Post-build event

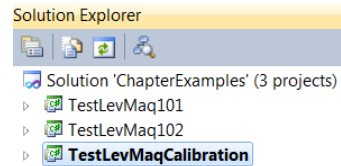
Figure 8: Excel-DNA needed files

3 Using the material in CsForFinancialMarketsLight directory

1. Open the directory `CsForFinancialMarketsLight` and chose a chapter, for example `Appendix2`
2. If the selected chapter includes many sub-directories as in Figure 9a, then there is a solution containing all projects of the chapter. In the example double-click the file `ChapterExamples.sln`. Visual Studio will start and in the `Solution Explorer` window you can see all contained projects as in Figure 9b.



(a) Sub-directories for a chapter



(b) Many projects in one solution

Figure 9: Example of chapter with many projects

3. Select the chosen project and check that references are ok, as explained in Section 2, point 3. In this example reference to `alglibnet2.dll` is needed
4. In the `Solution Explorer` window you can see the source code, organized in folders and sub-folders. Double-click to explode, see Figure 10.

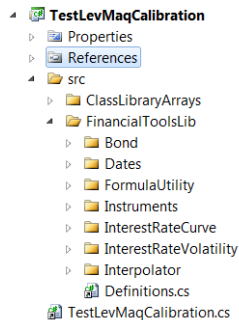


Figure 10: Source code files are organized in folders and added as links

Note that the code files are linked to the project (see 'How to: Add Existing Items to a Project' [http://msdn.microsoft.com/en-us/library/ee264131\(VS.100\).aspx](http://msdn.microsoft.com/en-us/library/ee264131(VS.100).aspx))

5. Right-click on the project, **Set as StratUp Project** and run it (`Ctrl+F5`).
6. Alternatively, to avoid loading unneeded projects in the chapter, you can open only the chosen project. In the example, open the directory `CsForFinanicialMarketLight/Appendix2/TestLevMqCalibration`, double-click on the file `TestLevMqCalibration.sln`. Finally, run the project (`Ctrl+F5`).