|  |  |
| --- | --- |
|  | Microsoft Biology Foundation Onboarding Guide  Version 1.0 - June 2010 |

Abstract

The Microsoft® Biology Foundation (MBF) is a language-neutral bioinformatics toolkit, built as an extension to the Microsoft .NET Framework. MBF is available under an open source license.

This document describes the steps for onboarding to the open source project so that you can contribute code to the Microsoft Biology Foundation.

For updates to this document and the rest of the MBF documentation, see   
<http://mbf.codeplex.com/documentation>.

For updates to MBF, see <http://mbf.codeplex.com>.

Contents

[Introduction 3](#_Toc264993451)

[Set Up Your Environment 3](#_Toc264993452)

[Installing Visual Studio 2010 4](#_Toc264993453)

[Getting MBF Source Code by Using the CodePlex TFS Server 4](#_Toc264993454)

[Building the Code 10](#_Toc264993455)

[Testing the Code 15](#_Toc264993456)

[How to Build the API Reference CHM 16](#_Toc264993457)

[Quick Start 17](#_Toc264993458)

[Connecting to the Server or an Existing Project 17](#_Toc264993459)

[Creating a New Work Item in Team Explorer 18](#_Toc264993460)

[Querying Work Items 18](#_Toc264993461)

[Changing Work Item Type 18](#_Toc264993462)

[Moving a Work Item to a Different Project 19](#_Toc264993463)

[Using Excel as a Front End 19](#_Toc264993464)

[Using Microsoft Project as a Front End 20](#_Toc264993465)

[Building Biology Extension Source for Microsoft Office 2010 20](#_Toc264993466)

[Conclusion 21](#_Toc264993467)

Disclaimer: This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Azure, MSDN, Silverlight, Visual Studio, and Windows are trademarks of the Microsoft group of companies.  All other trademarks are property of their respective owners.

# Introduction

Welcome to the Microsoft Biology Foundation (MBF). This document will get you up and running with the development practices and coding standards used. Your first steps will be to:

* Get ‘wired’ into the team by joining discussion lists and email notifications, locating relevant web sites, and browsing the documentation.
* Install the necessary tools.
* Create and configure your development environment.

We will cover all these steps in the following sections.

**Note**  This is a living document. If you find mistakes or areas that are confusing as you go through the process of getting setup, please log a work item to update the document. Thanks!

CAUTION This document assumes that you have been granted contribution rights to the CodePlex project. If you have not, the steps provided to synchronize to the source code via Visual Studio will likely fail due to lack of sufficient privileges. If you are interested in contributing, please see the Contribution Guide in the documentation section of the MBF project site.

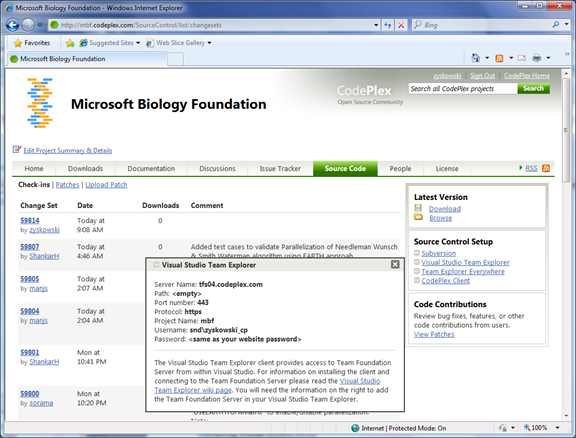
# Set Up Your Environment

The MBF project on CodePlex uses Microsoft® Visual® Studio Team Foundation Server (TFS) to manage the development process. Team Foundation Server includes collaboration components that support team development, such as source code control and work item/bug tracking.

Although several different Source Control programs can connect to the TFS repository on CodePlex, the information in this document is tailored for use of the Microsoft Visual Studio toolset. If you prefer to use one of the other supported source control tools, please see the documentation on CodePlex on how to connect with the source code.

Visual Studio includes most of the desktop development tools that you will need to work with MBF within an Integrated Development Environment (IDE). Such tools include program editors, compilers, linkers, and debuggers. Visual Studio is a retail product from Microsoft that runs on top of Windows, and can be obtained for free if you are in Academia via the [DreamSpark](https://www.dreamspark.com/) program.

The server that stores our source code is managed and maintained by CodePlex. Our current server is named tfs04.codeplex.com, but you should check the CodePlex site to make sure that the server has not changed. The following figure shows how to find the server information on the CodePlex site.

   
How to find the TFS server information for a CodePlex project

## Installing Visual Studio 2010

You should already have a system that you can logon, can obtain administrative rights to, and the ‘normal’ software installed— for example, Windows, Internet Explorer, Microsoft Office.

To install Visual Studio and work with TFS

1. Install Visual Studio.

2. Install Visual Studio Team Explorer.

These two steps install:

* The source control client programs and integration plug-ins that enable the Visual Studio IDE to communicate with TFS.
* The command line tools, so that you can interact with TFS from the ‘DOS’ box.

Remember that all of these tools are available for free via the [DreamSpark](https://www.dreamspark.com/) program for academic researchers.

For more information on Team Foundation Server, see [Getting Started with Team Foundation](http://msdn2.microsoft.com/en-us/library/ms181301(vs.90).aspx) on MSDN.

## Getting MBF Source Code by Using the CodePlex TFS Server

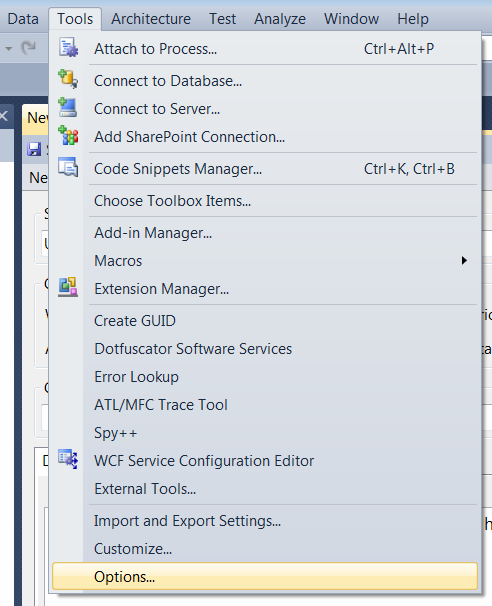
To obtain the current MBF source code, you must select Visual Studio Team Foundation Server as your source control plug-in and then download the latest version of the MBF source code from the CodePlex TFS server. If Visual Studio is properly installed as described in the previous section, these steps will get you connected to TFS.

**Note**. Issues with Security Groups might interfere with getting access to the source code by using Visual Studio/TFS. On occasion, a user’s first two or three attempts might fail to connect to CodePlex via Visual Studio, with an “invalid password/user account” message. Keep trying. After a few attempts your CodePlex credentials will be accepted. If not, contact CodePlex Support for help.

To connect to TFS

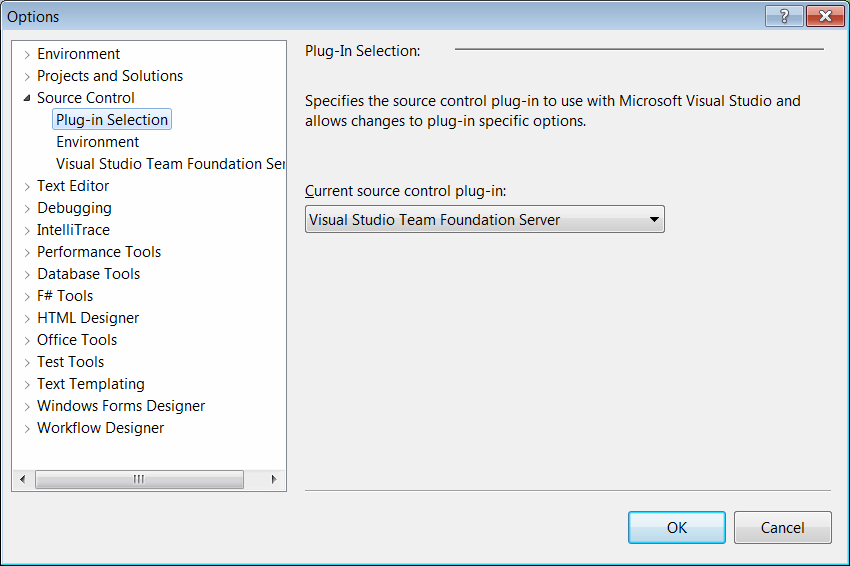
1. Start Visual Studio.

2. On the **Tools** menu, click **Options** to open the **Options** dialog box.



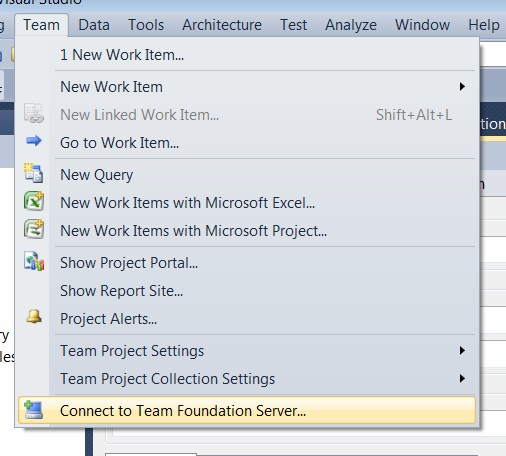
3. In the left panel of the **Options** dialog box, expand the **Source Control** item and click **Plug-in Selection.**

In the right panel, set **Current source control plug-in:** to Visual Studio Team Foundation Server. Click **OK**.

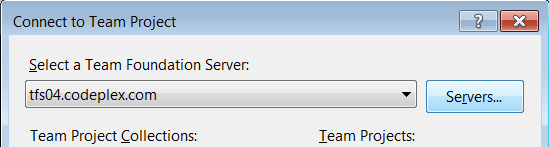


Now that TFS is your source control plug-in for the IDE, you can get the source code.

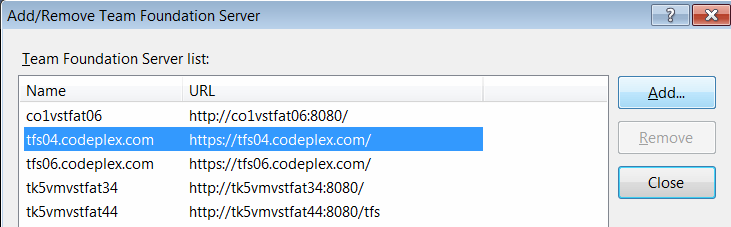
4. On the **Team** menu, select **Connect to Team Foundation Server.**



5. In the **Connect to Team Foundation Server** dialog Box, click **Servers**.



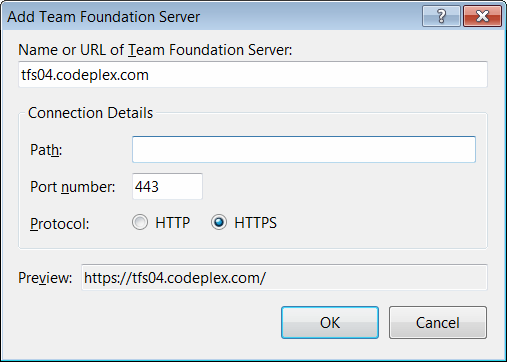
6. In the **Add/Remove Team Foundation Server** dialog box, click **Add**.



7. In the **Add Team Foundation Server** dialog box:

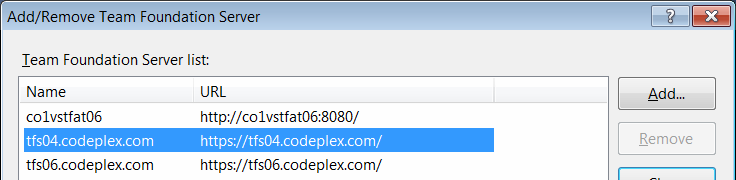
* Enter the name of the TFS server that hosts the project—tfs04.codeplex.com.
* Leave the Path field empty.
* Ensurethat the Port number is set to 443 and the HTTPS protocol is selected.

Then click **OK**.

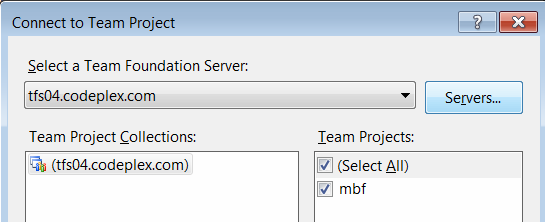


The tfs04.codeplex.com server should now appear in the list of Team Foundation Servers on your machine.

8. Click **Close** to return to the **Connect to Team Foundation Server** dialog box.

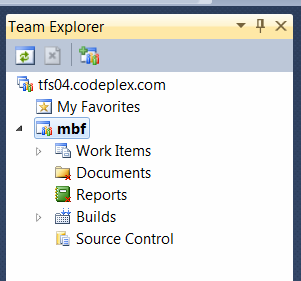


9. Make sure that the **MBF** Team Project is checked, and then click **Connect**.



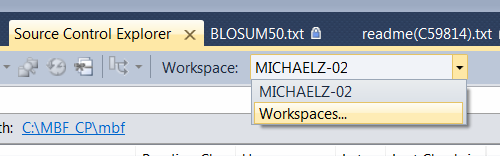
The project now appears in your **Team Explorer** window. If the **Team Explorer** window is not visible, press the key sequence **CTRL+\** followed by **CTRL+M** to activate it.

10. Double-click the **Source Control** tree item to launch the **Source Control Explorer** on the **MBF** project.



The next step is to create a **Workspace** that defines how you will work with the source on your machine.

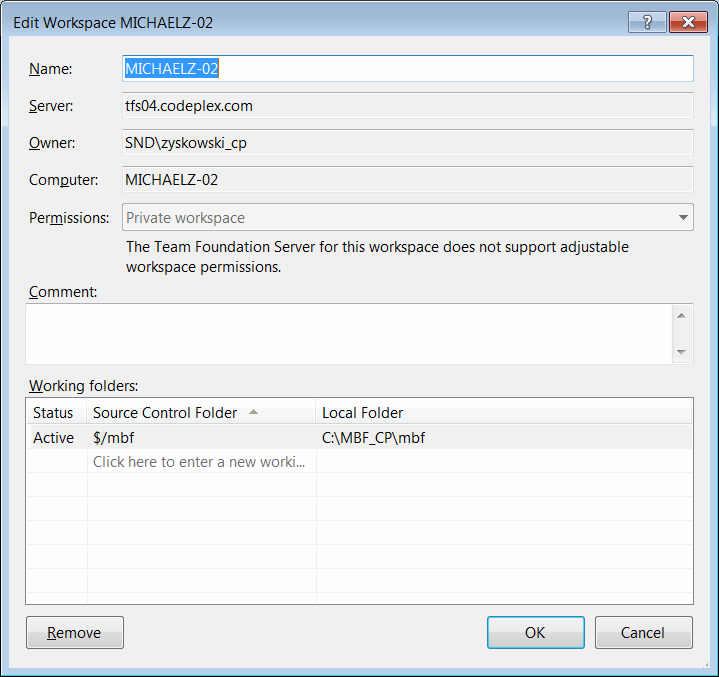
11. In the **Source Control Explorer** that you just opened, click Workspaces in the dropdown menu to display the **Manage Workspaces** dialog box.



The **Manage Workspaces** dialog is used to create your workspaces. Workspaces are the mechanism that tells TFS where on your system to put files that are under source control on so that you can work on them locally. Because you might have multiple enlistments in the source control system, you need to create a workspace name and tell TFS how to associate the files under source control on the server with directories and files on your local hard drive. You can create a local directory for the files if you select the B**rowse** […] button.

The complete MBF tree is small enough that we recommend that you map the entire project. A good working model is to have an MBF root directory on your machine and map the project there. After you have selected the projects you want, map the appropriate projects in source control to the appropriate directories in your local folder. If you do not map the entire project, keeping the path relationships the same is a *really* good idea, because it reduces confusion and some projects have tree assumptions built into them.

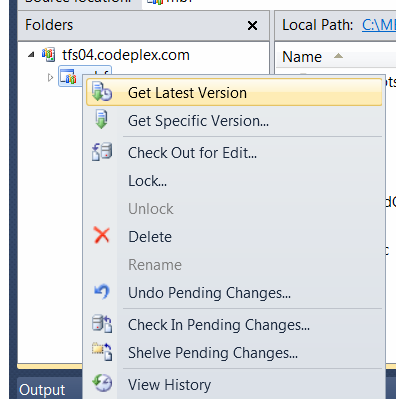
13. Once you have set your mappings appropriately, click **OK**.



Populating the workspace is the last step.

14. In the **Source Control Explorer** window, right-click the **MBF** project and click **Get Latest Version**.

This will retrieve the latest version of the source files from the source control server, and you will be able to work on the code.

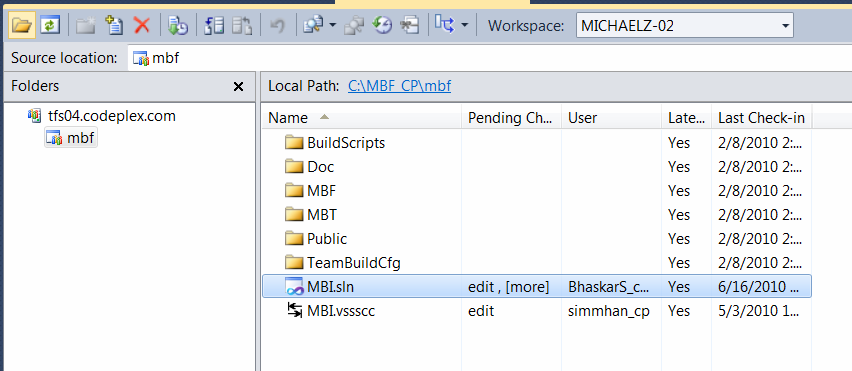


You should now have the MBF source code tree on your system and be able to navigate and build the tree. Source control is an important part of the development process and you will be working with TFS frequently. It is important that you become well acquainted with it.

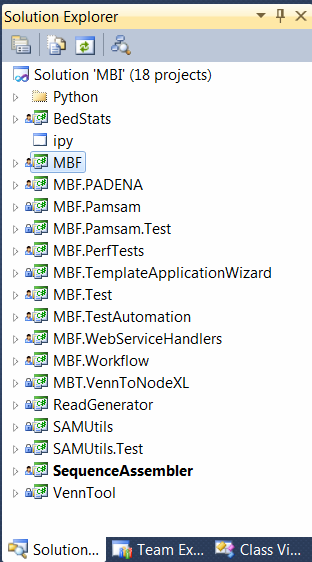
For more information on using TFS, see [Getting Started with Team Foundation](http://msdn2.microsoft.com/en-us/library/ms181301(vs.90).aspx) on MSDN.

## Building the Code

The first step in building MBF is to open the Solution file for MBI (Microsoft Biology Initiative), which is named MBI.sln. This file is in the MBF root folder .



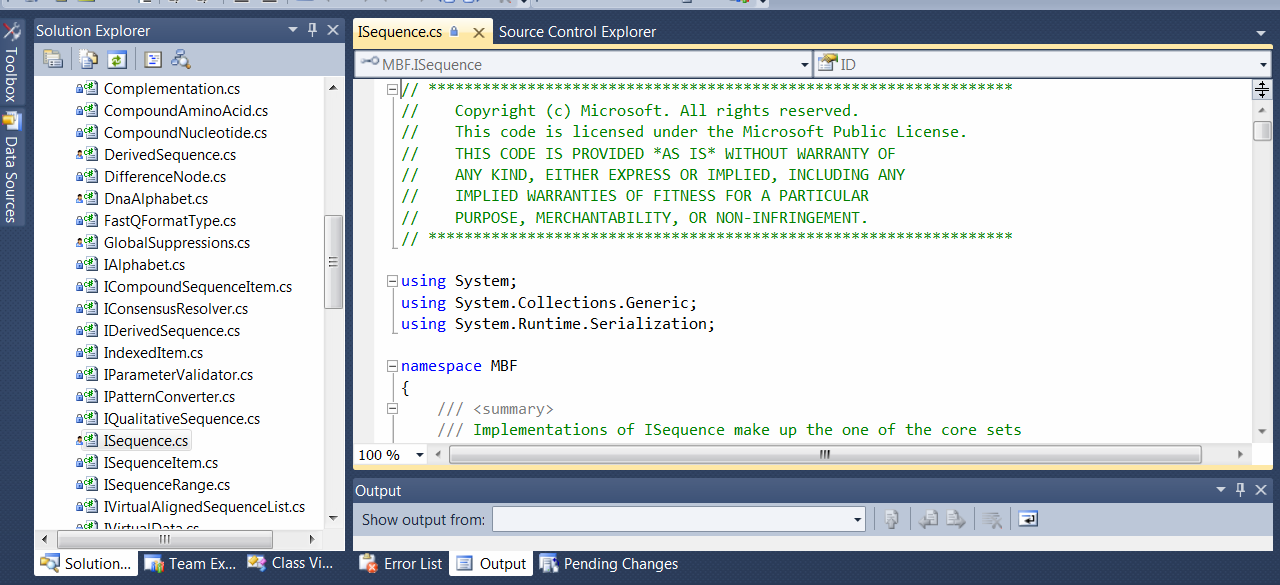
This will activate the contents of the **Solution Explorer**. To view the **Solution Explorer**, click the tab on the bottom left, next to the Team Explorer tab:



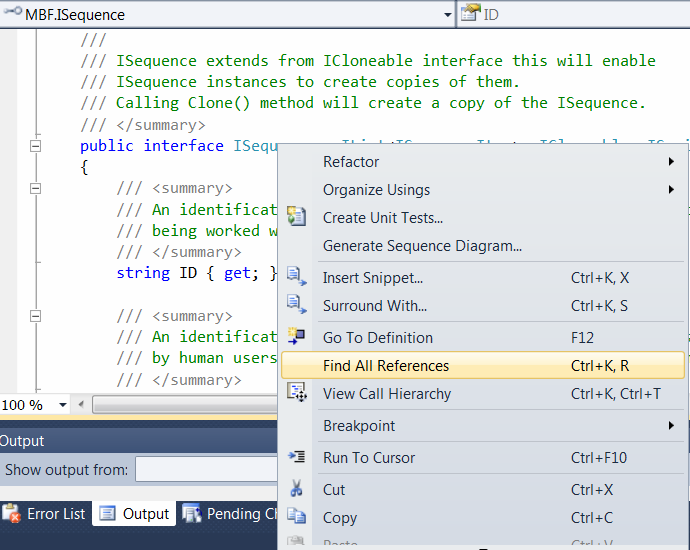
If the development machine does not have the Silverlight developer runtime installed, users will receive several popups asking to install it and SilverMap contains no items.

At this point, you can navigate, explore and review the code by expanding the folder structure and double-clicking on the source code files of interest.

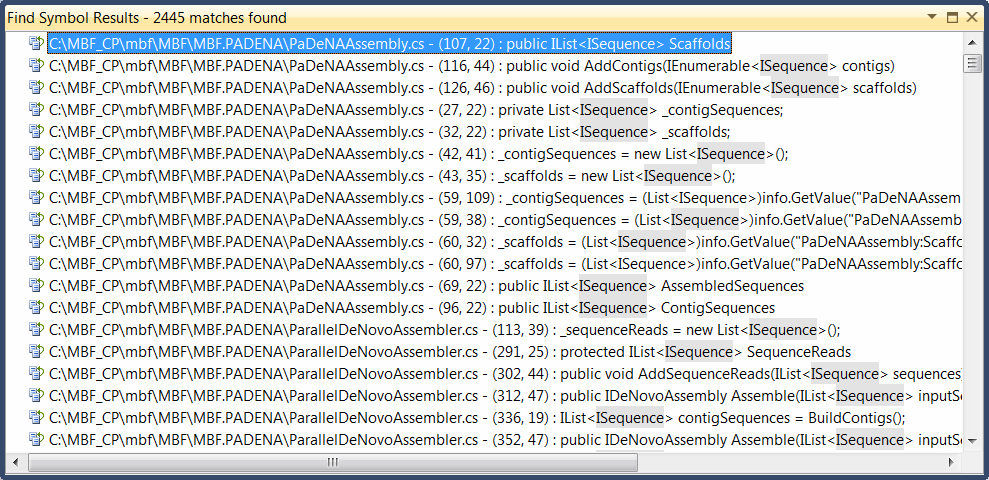
For example, if you expand the Bio component and select the ISequence.cs item, Solution Explorer displays output similar to the following :



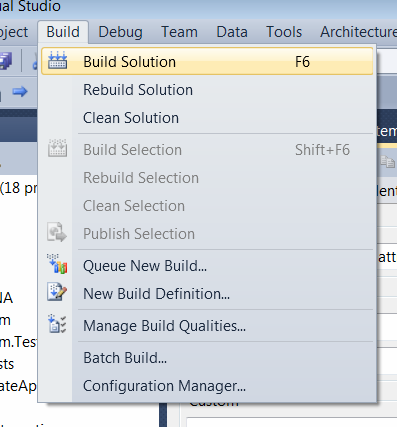
It’s easy to navigate the code this way, and Visual Studio provides quite a bit of navigational assistance through context menus. For example, if you right-click the term ISequence in the interface declaration, a pop-up menu provides the **Find All References** option.



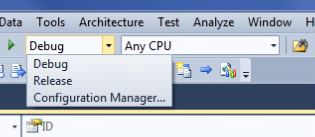
You can then jump to areas of the code that use this interface, providing a bit more context and examples of how it can be used in your own code:



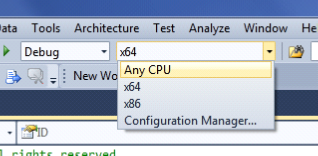
To build the application, open the **Build** menu and click **Build Solution**:



Choose whether to build a Release or Debug version of the code in the drop-down menu in the toolbar:



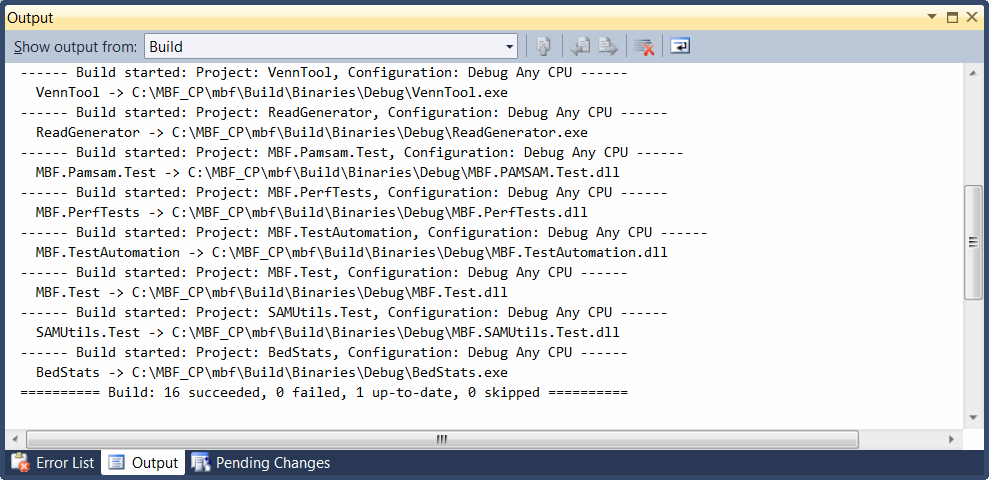
Similarly, choose the target build environment in the other dropdown menu:



To monitor the progress of the build, we suggest that you enable the **Output** windowby selecting **Output** from the **View** menu:



When the build is complete, the Output window shows the following information:



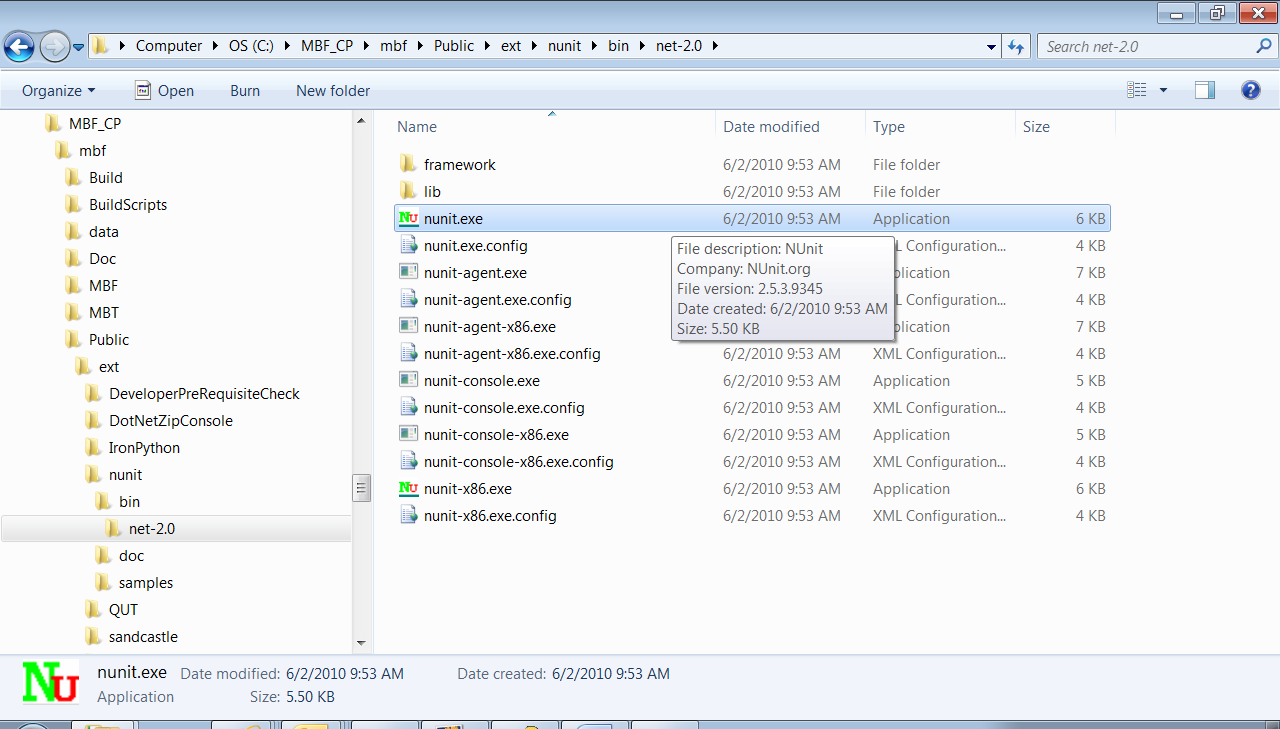
## Testing the Code

The primary means for ensuring code quality in MBF is the use of the NUnit Framework (<http://www.nunit.org>). NUnit is another open-source project that is designed to make it easy to develop, integrate, and run test code against your application.

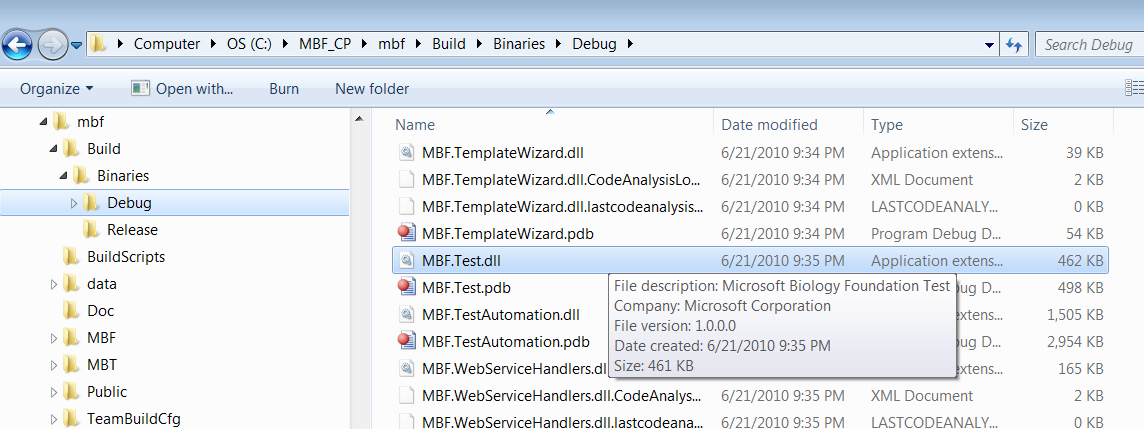
As part of the MBF source code, NUnit tests are provided to exercise each interface, function, and feature from both unit test and automated perspective. The tests can be found in the MBF.Test and MBF.TestAutomation trees.

Important If you develop any new code, or extension to existing code, you must include a new NUnit test when you submit the changes. This ensures that subsequent development does not inadvertently cause a breaking change to another part of the code base. It also informs others about the intent of a particular piece of code—thus helpingpeople who are new to the project to come up to speed on areas of interest.

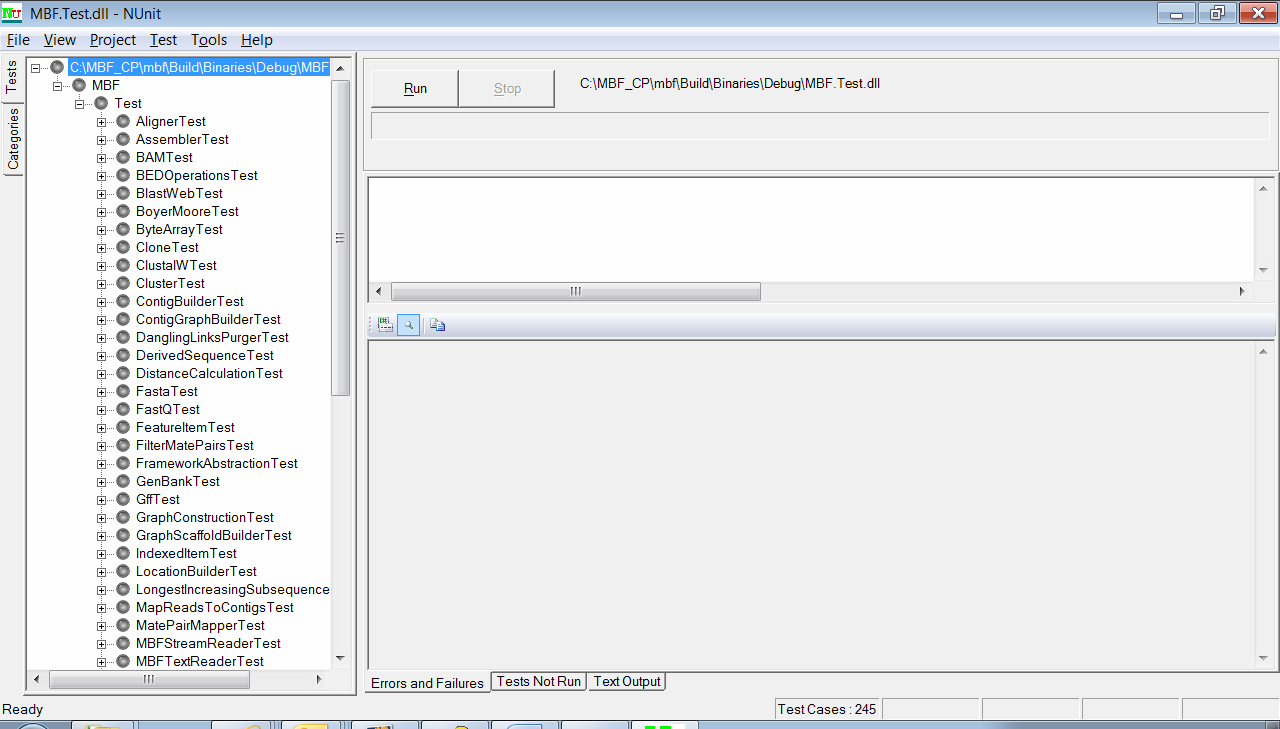
To run the tests, download and install the NUnit Framework and start the NUnit GUI:



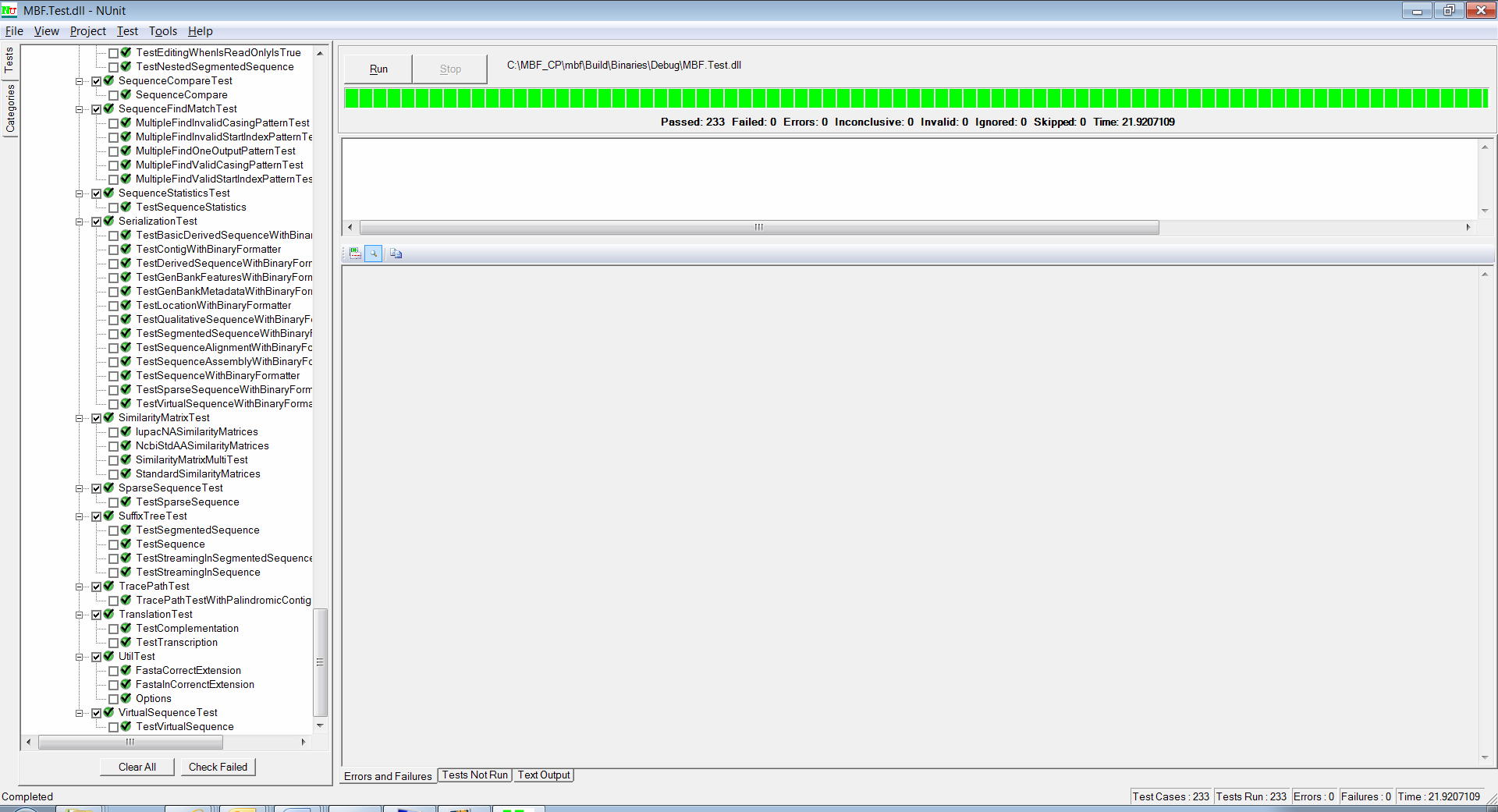
After the GUI starts, click **Open** on the File menu and then navigate to the mbf\Build\Binaries\Debug directory. Click the MBF.Test.dll file that your local build generated:



This opens a list of the NUnit test cases that the code produced:



If you run the tests, you should see something like this:



## How to Build the API Reference CHM

To build the MBF API Reference help file, use the current version of the Sandcastle Help File Builder from the Codeplex site at <http://shfb.codeplex.com/>.

See \BuildScripts\BuildScripts\_ReadMe.docx for additional information.

To prepareSandcastle

* Install Sandcastle from <http://shfb.codeplex.com/>.
* Copy the \public\ext\sandcastle\ProductionTools\MrefBuilder.exe.config file to the Sandcastle installation folder on your machine.
* Copy the config file from \public\ext\wix3 to the Sandcastle installation folder.
* From a command prompt, run \BuildScripts\DeveloperPreRequisiteCheck.exe.

DeveloperPreRequisiteCheck.exe sets the required environment variables.

* Enter the Sandcastle install folder.
* Enter the path to the HTML Help Compiler.
* Set the %SOURCEROOT% environment variable to c:\ProjectFolderName\mbf.

**Note I**f you accept the defaults, the root folder is actually mbf, which is placed under the folder that you mapped the project to.

* Set the %BINARYROOT% environment variable to c:\<YourProjectFolderName>\mbf\Build\Binaries.
* Set the %SANDCASTLE\_INSTALLPATH% environment variable to the SandCastle installation folder.
* Set the %HHC\_INSTALLPATH% environment variable to the HTML Help Workshop installation folder.

To build a CHM file with Sandcastle

* Run mbf\BuildScripts\DocForTFS.cmd.

The CHM is created in the C:\MBF\_Source\mbf\Build\Binaries\Debug\Doctemp\Doc\Output folder.

**Note**  If you map Team Server to a folder on your hard drive, the folder name can not have spaces.

# Quick Start

<You need an introductory sentence here: who is this for, how does it differ from everything else. Also, if this is intended for somewhat experienced people, suggest mentioning it early on so that they can skip the rest.>

## Connecting to the Server or an Existing Project

To connect to the server, specify the name of the application tier of the Visual Studio Team Server (VSTS) installation.

1. Start the Visual Studio IDE.

2. On the Tools menu, click Connect to Team Foundation Server.

3. Click Servers and then click Add on the Connect to Team Project dialog box.

4. Type the name of the application tier and the port to connect to.

5. Click **OK**, and then click Close.

6. Choose the server that you just added in the drop-down menu.

7. Click the projects you want to connect to, and then click **OK**.

## Creating a New Work Item in Team Explorer

To add a new work item, you need to know which project to add the bug to. Connect to that project, and then do the following:

1. Expand the project and right-click the Work Items folder.

2. Expand **Add Work Item** and choose the type of work item type to add.

3. Enter information in the yellow fields.

4. Save the work item by clicking **Save** on the File menu orby pressing CTRL+S.

## Querying Work Items

You can view, modify, and add work item queries by using VSTS. When you query work items, VSTS displays only the project hierarchy, and not the path hierarchy. You can query from any project to another, but if you know which project you want to query, connect to that project.

To view an existing query

1. Expand the Work Items folder and choose a query in the Team Queries or My Queries folder.

2. Double-click the query to view the results of that query.

To modify a query

1. Access the query clauses under View > Query.

2. Choose new query values from the drop-down lists or type in new values as appropriate.

3. To group clauses, go to Team > Clauses > Group Clauses.

To add a new query

1. Right-click the Work Items, Team Queries, or My Queries folders.

2. Choose New Query.

3. Follow the steps in the previous procedure to add clauses and supply query values.

## Changing Work Item Type

There is currently no support for changing a work item type. Instead, you create a copy of the work item that has the new type.

To change the type of a work item

1. Open the work item or select it in the query results list.

2. On the Edit menu, click Create Copy of Work Item.

3. Choose the project to copy the work item to. To change the type of the item, choose the current project.

4. Choose the type of work item for the copy.

5. Update and enter data as required in all yellow fields.

6. Save the copied work item by clicking **Save** on the File menu or by pressing CTRL+S.

7. Close the original work item. The new work item will contain a link to the original work item.

## Moving a Work Item to a Different Project

There is currently no support for moving a work item to a different project. Instead, you create a copy of the work item in the other project. Follow the steps in the previous section for changing the type of a work item. In step 3, select the project to which you want to move the item.

## Using Excel as a Front End

You can add new work items and update or view existing work items by using Microsoft Excel 2007 or 2010. There are a couple of ways to connect using Excel:

* Right-click a work item in the query results list and click Open Selection in Microsoft Excel.
* Start Excel, click New List on the **Team** menu, and select the server and project connect to. Then choose one of the following:

Query **List** to view work items from an existing query.

Input List to create new items or get a specific item.

Here are steps for some common actions in Excel.

|  |  |
| --- | --- |
| To perform this action: | Use this menu item: |
| Get a list of work items | On the **Team** menu, click **Get Work Items** |
| Save new work items or updates in the VSTS database | On the **Team** menu, click **Publish Changes** |
| Refresh data | On the **Team** menu, click **Refresh** |
| Change the work items viewed | On the **Team** menu, click **Configure List** |
| Change the columns displayed | On the **Team** menu, click  **Choose Columns** |
| Save formatting in the Excel document for later viewing | On the **File** menu, click **Save,** or press Ctrl+S |
| If you want to save the work item to the project’s document store, save it to http://<server name>/sites/<project name>/<document library name>/<any folder names>. | |

## Using Microsoft Project as a Front End

You can add new work items and update or view existing work items by using Microsoft Project®. There are a couple of ways to connect using Project:

* Right-click a work item in the query results list and click Open Selection in Microsoft Project.
* Start Project, click Choose Team Project on the Team menu, and select the server and project to connect to.

Here are steps for some common actions in Project.

|  |  |
| --- | --- |
| To perform this action: | Use this menu item: |
| Get or change the list of work items | On the **Team** menu, click **Get Work Items** |
| Save new work items or updates in the VSTS database | On the **Team** menu, click **Publish Changes** |
| Refresh data | On the **Team** menu, click **Refresh** |
| Save formatting in the Project document for later viewing of this set of work items | On the **Team** menu, click **Configure List** |
| If you want to save the work item to the project’s document store, save it to http://<server name>/sites/<project name>/<document library name>/<any folder names>. | |

## Building Biology Extension Source for Microsoft Office 2010

If you use Microsoft Office 2010, follow these steps to convert the Microsoft Research Biology Extension for Excel.

To convert the Biology Extension source

1. In Visual Studio 2010, open the BioExcel project, which launches the conversion wizard.

2. Complete the conversion wizard.

**Recommended** When prompted by the wizard, choose to create a backup.

When the wizard completes successfully, a few unwanted references are added to the project, because BioExcel is designed to be built against the Office 2007 runtime. The following steps clean up the references.

To remove unwanted references for BioExcel

1. In Visual Studio, remove the following references from the BioExcel project:

Microsoft.Office.Tools.Common.v9.0.dll  
Microsoft.Office.Tools.Excel.v9.0  
Microsoft.Office.Tools.v9.0  
Microsoft.VisualStudio.Tools.Applications.Runtime.v9.0  
System.AddIn  
Microsoft.Office.Interop.Excel (v 14.x.x.x)

2. Add the following references:

Microsoft.Office.Tools.Common  
Microsoft.Office.Interop.Excel (v 12.x.x.x)

After you make these changes, the BioExcel project will build without errors, but with a few warning messages. Follow these additional steps to fix the warnings.

To remove the cause of warning messages

1. In **Solution Explorer**, right-click the BioExcel project node.

2. Click Unload Project.

3. Right-click the BioExcel project node again, and click Edit BioExcel.

4. In the project file, find two entries for the following line:

<Import Project="$(MSBuildExtensionsPath)\Microsoft\VisualStudio\v10.0\OfficeTools\Microsoft.VisualStudio.Tools.Office.targets" />

5. Delete those two entries and save the project.

Note After you complete these steps, you might receive one last message that the Office 2007 PIA is missing. You can safely ignore this warning. The PIA is the Primary Interop Assembles redistributable package, which is not necessary to run the Biology Extension. However, you can install the PIA from the following link—but only if you have Office 2007 installed:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=59daebaa-bed4-4282-a28c-b864d8bfa513>

To complete the conversion for Office 2010

* Set BioExcel as a startup project, and you are ready to press F5.

# Conclusion

This document was intended to provide a quick introduction to get you started developing for MBF. Please make sure you become very familiar with the MBF Coding Guidelines document (MBF\_Coding\_Conventions.docx) as well gain a thorough understanding of how the various parts interact, before you attempt to modify the code base. Having a good background on the project will help to make sure your first code review goes smoothly.

Welcome to MBF!