|  |  |
| --- | --- |
| appicon3_Large.jpg | .NET Bio Onboarding Guide |

Version 1.1 – June, 2013

Abstract

This document describes

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

For updates to .NET Bio, see <http://bio.codeplex.com>.

# Contents

[Contents 2](#_Toc312054230)

[Introduction 4](#_Toc312054231)

[Set Up Your Environment 4](#_Toc312054232)

[Installing Visual Studio 2010 4](#_Toc312054233)

[Getting Source Code 5](#_Toc312054234)

[Building the Code 11](#_Toc312054235)

[Testing the Code 19](#_Toc312054236)

[To run the tests 21](#_Toc312054237)

[Conclusion 21](#_Toc312054238)

[Quick Start: 22](#_Toc312054239)

[Connecting to the Server or an Existing Project 22](#_Toc312054240)

[Creating a New Work Item in Team Explorer 22](#_Toc312054241)

[Querying Work Items 22](#_Toc312054242)

[Changing Work Item Type 23](#_Toc312054243)

[Moving a Work Item to a Different Project 23](#_Toc312054244)

[Using Excel as a Front End 23](#_Toc312054245)

[Using Project as a Front End 24](#_Toc312054246)

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

# Introduction

This document will get you up and running with the development practices and coding standards used in the .NET Bio project. 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 doc. Thanks!

# Set Up Your Environment

The .NET Bio project on CodePlex uses Visual Studio Team Foundation Server (TFS) to manage the development process. Team Foundation Server includes the collaboration components that support team development. This includes tools such as source code control, work item/bug tracking, and document management among others.

There are a few different Source Control programs which can connect to the TFS repository on CodePlex, but the information provided in this document is tailored for use of the Microsoft Visual Studio toolset. If you wish 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 needed desktop development tools within an Integrated Development Environment (IDE). This includes tools such as program editors, compilers, linkers, and debuggers. This 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 server is named **tfs09.codeplex.com**, but you should check the CodePlex site to make sure the server listed hasn’t changed.

## Installing Visual Studio

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

To install Visual Studio and work with TFS

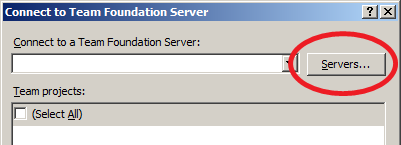
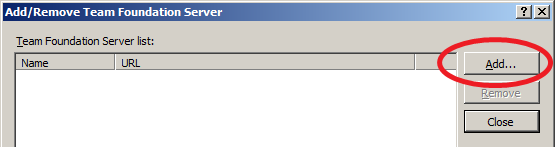
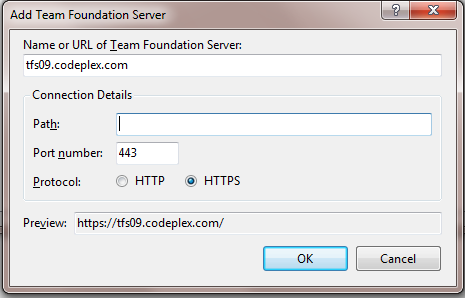
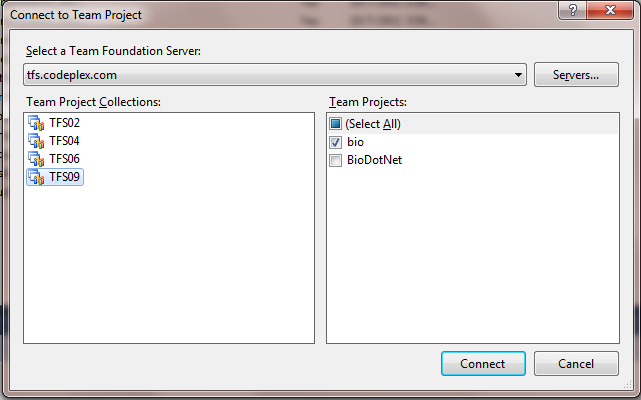
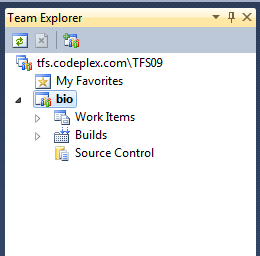
1. Install Visual Studio 2012 or 2012.
2. Install Visual Studio Team Explorer.

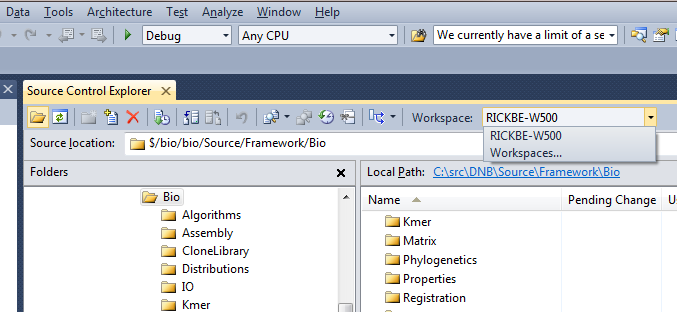
This installs the source control client programs and integration plug-ins that enable the Visual Studio IDE to talk with TFS. It also installs the command line tools so you can interact with TFS from the ‘DOS’ box.

For more information on Team Foundation, see Appendix 1, or refer to the MSDN article: [Getting started with Team Foundation](http://msdn2.microsoft.com/en-us/library/ms181301(vs.90).aspx)

## Getting Source Code

To get the current source code on your machine to work with, you must select Visual Studio Team Foundation Server as your source control plug-in and then download the latest version from the CodePlex TFS server. Presuming you have Visual Studio properly installed from the previous section, these steps will get you connected to TFS. To connect to TFS

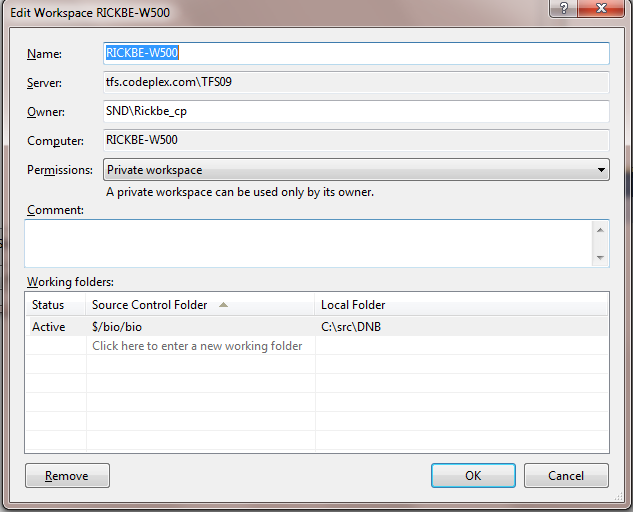
1. Start Visual Studio.
2. On the **Tools** menu, select ***Options*** which will open the **Options** dialog
3. In the left panel of the **Options** dialog box, expand the **Source Control** item and select **Plug-in Selection.** In the right panel set the **Current source control plug-in:** to Vi**sual Studio Team Foundation Server** and click ***OK***.  
   
4. Now TFS is your source control plug-in for the IDE, let’s get the source code. On the **Team** menu, select ***Connect to Team Foundation Server*…  
   **
5. The **Connect to Team Foundation Server** dialog appears asking for the server to connect to. Click the ***Servers****…* button  
   
6. This brings up the **Add/Remove Team Foundation Server** Dialog. Click the ***Add*** button   
   
7. The **Add Team Foundation Server** dialog should now be active. Enter the name of the TFS server that hosts the project (**tfs09.codeplex.com**) and ensure the Port number is set to **443** and the **HTTPS** protocol is selected. Then click the ***OK*** button.   
     
   
8. You should now have the **tfs09.codeplex.com** server in the list of Team Foundation Servers your machine is aware of. Click the ***Close*** button to return to the **Connect to Team Foundation Server** dialog.  
     
   
9. Make sure that the.NET BioTeam Project is checked, and then click ***Connect***.  
     
   
10. You should now have the project visible in your **Team Explorer** window. (If the **Team Explorer** window is not visible, the sequence “***Ctrl+\ Ctrl+M***” should activate it.) Double clicking on the **Source Control** tree item will launch the **Source Control Explorer** on the **.NET Bio** project.   
    

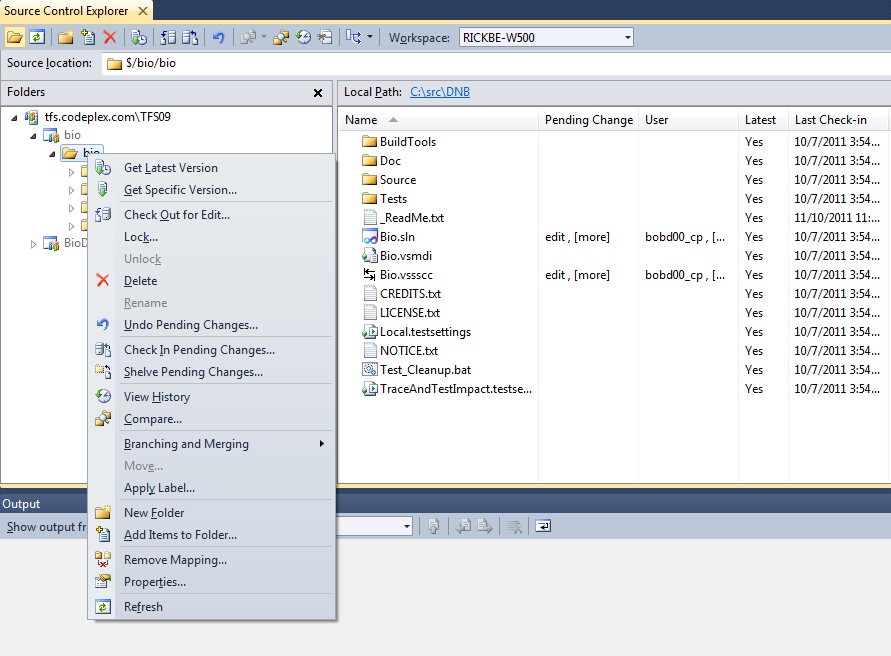
The next step is to create a **Workspace** that defines how you will work with the source on your machine. This is done in the **Source Control Explorer** that you just opened. Clicking on Workspaces in the dropdown will bring up the **Manage Workspace** dialog   


1. The **Manage Workspaces** dialog is used to create different workspaces for you. Workspace are the mechanism that tells TFS where to put files that are under source control on your system so allowing you to work on them locally. Since you may 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 the local directory if you select the browse […] button.

The complete .NET Bio tree is small enough that we recommend that you map the entire project. A good working model is to have a .NET Bio root directory on your machine and map the project there. Once 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 as it reduces confusion and some projects have tree assumptions built into them.

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

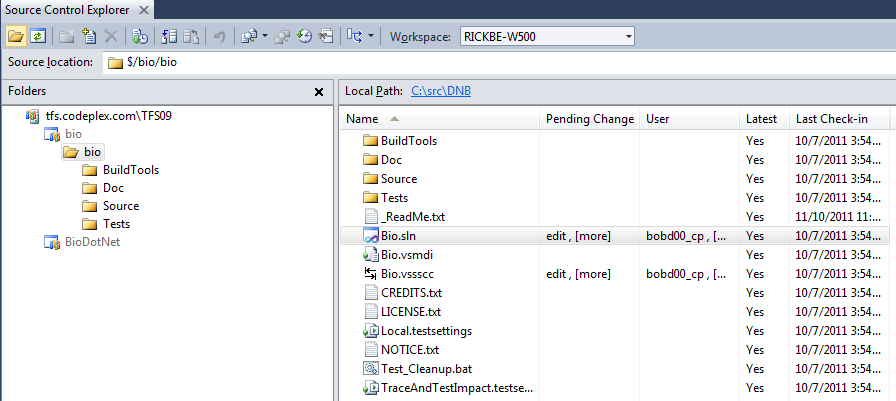


1. Populating the workspace is the last step. In the Source Control Explorer window, right click on the **.NET Bio** project and select ***Get Latest Version***. This will retrieve the latest version of the source files from the source control server and you will be able to work on the code.   
   
2. For more information on using TFS see the appendix

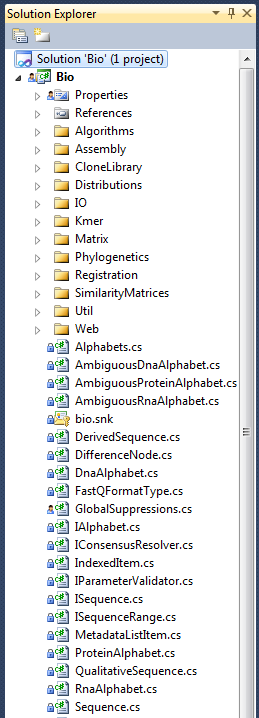
You should now have the .NET Bio source code tree on your system and be able to navigate and build the tree. Source control is an important part of the development process and you will be constantly working with TFS. It is important that you become well acquainted with it and to help you in that, there are more links to TFS training in the Appendix.

## Building the Code

The first step is to open the Solution file for .NET Bio, **Bio.sln**, alternatively to build just the core libraries and unit tests, use the **Bio.Core.sln** solution.

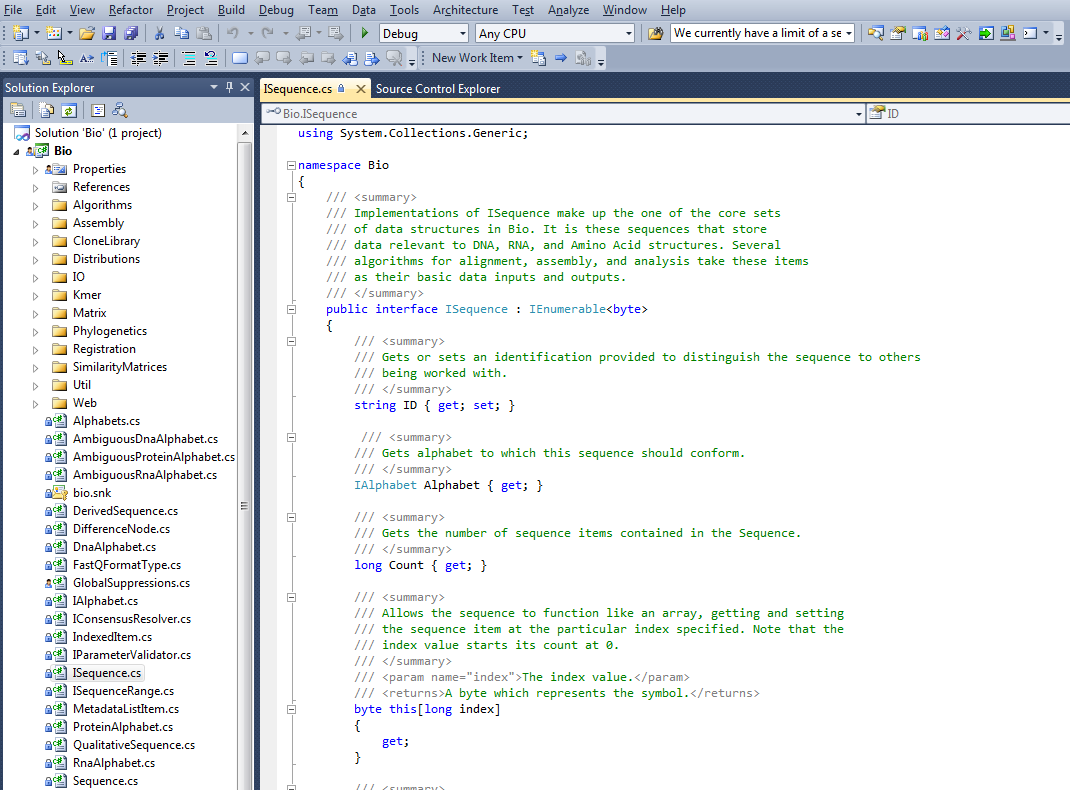


This will activate the contents of the Solution Explorer, which can be seen by clicking on the tab on the bottom next to the Team Explorer tab:

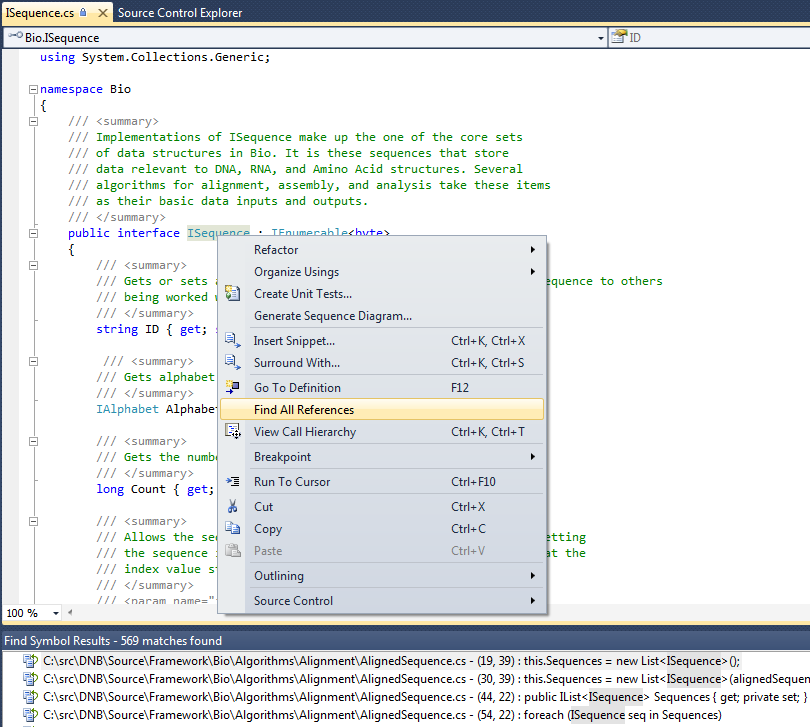


At this point, you can navigate, explore and review the code by expanding the folder structure and double-clicking on the code files of interest.

For instance, when expanding the Bio component and selecting the ISequence item:

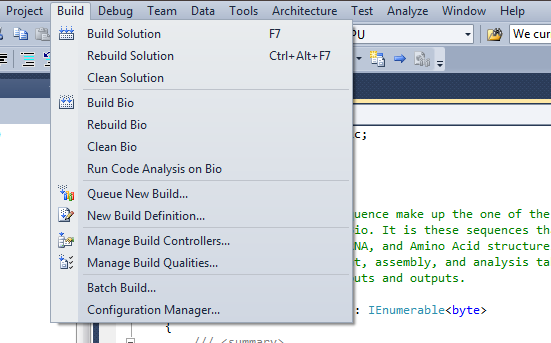


It’s quite simple to navigate the code this way, and Visual Studio in fact gives quite a bit of nice navigational assistance via context menus. For instance, right click on the term **ISequence** in the interface declaration, and a menu provides the option to “Find All References…”

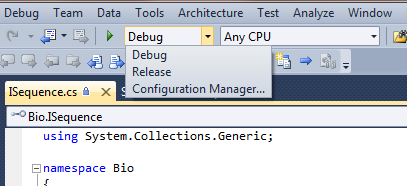


You then can navigate by jumping to areas of the code that use this interface, providing a bit more context and examples of how it can be used in your own code:

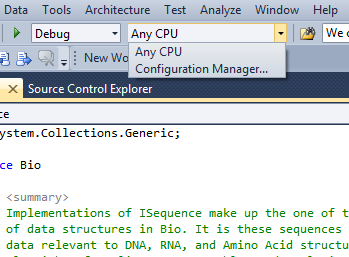
To build the application, simply select the Build toolbar at the top and Build Solution:



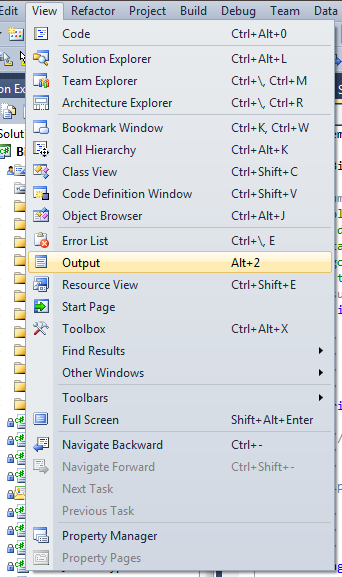
The option to build Release or Debug versions of the code are set via the drop down in the top toolbar:



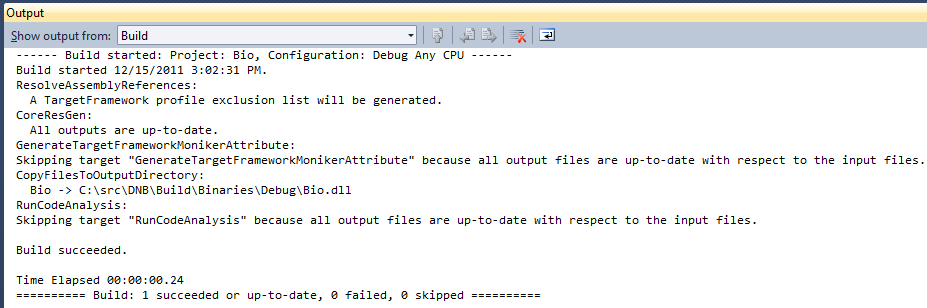
And similarly, selection of different target build environments are also possible via the other dropdown:



To monitor the progress of the build, I like to have the Output window enabled. This can be done by selecting the View option and then selecting the Output option:



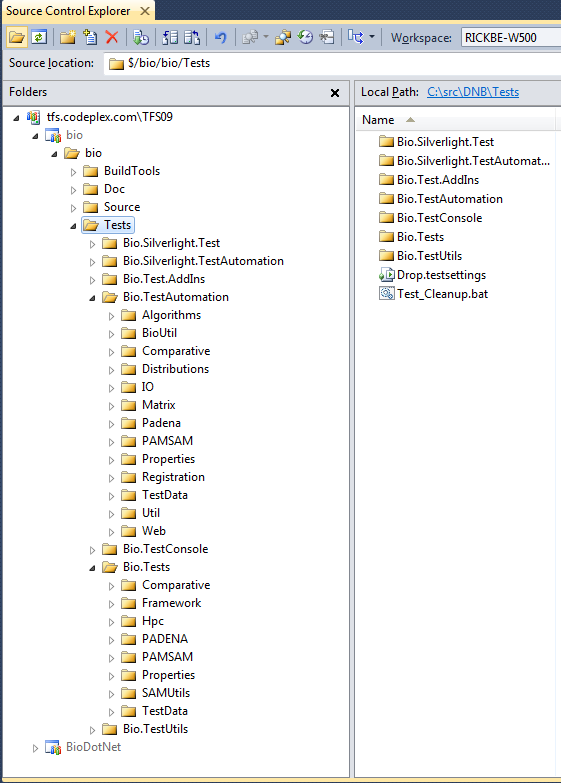
Hopefully, this is what you’ll then see when the build is completed:



## Testing the Code

The primary means for ensuring code quality in .NET Bio is the use of MSTest, part of the Visual Studio Team System suite which includes a full test management system in TFS.

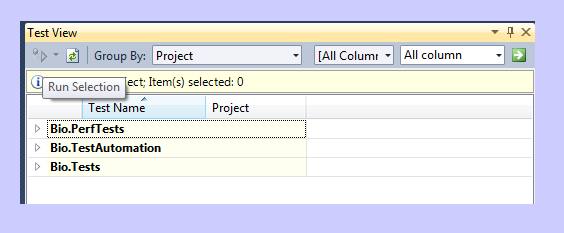
See the .NET Bio testing guide for specifics. The tests can be found in the Bio.Test and Bio.TestAutomation trees:



The development of any new code, or extension to existing code, MUST include a new unit test to be submitted with the changes. This ensures that subsequent development does not inadvertently cause a breaking change to another part of the code base, and is also helpful for educating others on the intent of a particular piece of code (hence aiding in the ability of people new to the project to come up to speed on areas of interest).

## To run the tests

1. Click **Windows** on the **Test** menu and open the **Test View** window.

2. In the **Test View** window select Bio.Tests and click **Run Selection**—the green arrow in the upper left corner—as shown in the following figure.

Test View window

# Conclusion

This document was intended to provide a quick introduction in how to get started developing for .NET Bio. Please make sure you become very familiar with the Coding Guidelines document, as well gain a thorough understanding of how the various parts interact, before attempting to modify the code base. Gaining a good background on the project will help to make sure your first code review goes smoothly.

Welcome to .NET Bio!

# 

# Quick Start:

## Connecting to the Server or an Existing Project

You’ll need to specify the name of the application tier of the VSTS install in order to connect.

1. Start the VS IDE.
2. Go to Tools->Connect to Team Foundation Server…
3. Click on Servers…->Add…
4. Type in the name of the application tier you wish to connect to, as well as the port.
5. Click OK, then Close.
6. Choose the server you’ve just added in the drop down.
7. Select the projects you want to connect to, then OK

## Creating a New Work Item in Team Explorer

To add a new Work Item, you’ll need to know which project you plan to add the bug to. Once you’ve connected to that project, do the following:

1. Expand the project and right click on the Work Items folder.
2. Expand Add Work Item and choose the Work Item type you’d like to add.
3. Enter information in the yellowed fields.
4. Save (File->Save, Ctrl-S).

## Querying Work Items

Path hierarchy is not displayed when querying work items in VSTS; only the project hierarchy. You can query from any project to another, but if you know which project you want to query, connect to that project. Then, do the following:

1. 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 to view.
2. To modify a query:
   1. Access the query clauses under View->Query.
   2. Choose values from the drop down lists or write in as appropriate.
   3. To group clauses, go to Team->Clauses->Group Clauses.
3. To add a new query:
   1. Right click on Work Items, Team Queries, or My Queries folders.
   2. Choose Add Query.
   3. Follow the steps under modify a query.

## Changing Work Item Type

There is currently no support for changing a work item type without creating a copy of the work item.

1. Open the work item or select it in the query results list.
2. Edit->Create Copy of Work Item…
3. Choose which project you want to copy the work item to.
4. Choose which work item type you’d like to copy it to.
5. Update/enter data in all yellowed fields.
6. Save.
7. Close the original work item. (There is a link to it in the new work item).

## Moving a Work Item to a Different Project

There is currently no support for moving a work item to a different project without creating a copy of the work item. The instructions here are the same as for changing a work item type.

## Using Excel as a Front End

You can add new work items and update view existing work items using Excel.

1. There are a couple of ways to connect using Excel:
   1. Right click on work item(s) selected in the query results list and choose Open Selection in Microsoft Excel…
   2. Open Excel and choose Team->New List and select the server and project you want to connect to. Then choose one of the following:
      1. Query for viewing work items from an existing query.
      2. Input list for creating new items or getting a specific item.
2. To get a list of work items: Team->Get Work Items.
3. To save new work items/updates to the VSTS database: Team->Publish Changes.
4. To refresh data: Team->Refresh.
5. To change the work items viewed: Team->Configure List.
6. To change the columns displayed: Team->Choose Columns.
7. To save formatting in the Excel doc for later viewing: Ctrl-S or File->Save. If you want to save it to the project’s document store, save it to http://<server name>/sites/<project name>/<document library name>/<any folder names>.

## Using Project as a Front End

You can add new work items and update view existing work items using Project.

1. There are a couple of ways to connect using Project:
   1. Right click on work item(s) selected in the query results list and choose Open Selection in Microsoft Project…
   2. Open Project and choose Team->Choose Team Project… and select the server and project you want to connect to
2. To get or change the list of work items: Team->Get Work Items.
3. To save new work items/updates to the VSTS database: Team->Publish Changes.
4. To refresh data: Team->Refresh.
5. To save formatting in the Project doc for later viewing of this set of work items: Ctrl-S or File->Save. If you want to save it to the project’s document store, save it to http://<server name>/sites/<project name>/<document library name>/<any folder names>.