|  |  |
| --- | --- |
|  | Code Contributor Guide  Version 2.0.Beta1 - April 2011 |

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 how to 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 to the contribution process 3](#_Toc289629574)

[About code contributions 4](#_Toc289629575)

[Contributions timelines and acceptance considerations 4](#_Toc289629576)

[The contribution process 5](#_Toc289629577)

[Contributor getting started 5](#_Toc289629578)

[Register at CodePlex and accept Terms of Contribution 5](#_Toc289629579)

[Sign up for MBF email and discussion groups 6](#_Toc289629580)

[Install TortoiseSVN 6](#_Toc289629581)

[Use available resources 6](#_Toc289629582)

[Dependent software products 7](#_Toc289629583)

[Review the C# coding and documentation guidelines 8](#_Toc289629584)

[Review existing code 8](#_Toc289629585)

[Create a new CodePlex work item 9](#_Toc289629586)

[Write the new code 9](#_Toc289629587)

[Write documentation for the code 10](#_Toc289629588)

[Write unit tests for the new code 10](#_Toc289629589)

[Run tests 10](#_Toc289629590)

[Run FxCop 10](#_Toc289629591)

[Prepare a file to submit 11](#_Toc289629592)

[Request code review 11](#_Toc289629593)

[Submit your changes on CodePlex 11](#_Toc289629594)

[Respond to code review recommendations 13](#_Toc289629595)

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

# Introduction to the contribution process

The Microsoft Biology Foundation (MBF) is a language-neutral bioinformatics toolkit, built as an extension to the Microsoft .NET Framework. MBF implements the following:

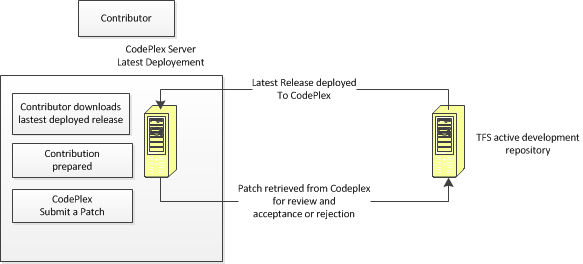
* An object model for representing genomic data.
* A range of parsers for common bioinformatics file formats.
* A range of algorithms for manipulating DNA, RNA, and protein sequences.
* A set of connectors to biological Web services such as NCBI BLAST.

MBF is available under an open source license. Executables, source code, demo applications, and documentation are freely downloadable from <http://research.microsoft.com/bio> for executables and training materials and from [http//mbf.codeplex.com](http://mbf.codeplex.com/) for source code and documentation.

MBF is available under an open source license. There are two avenues of participation in the project:

* As a Contributor – accessing the codebase deployed on Codeplex as self-contained .zip file and submitting your work through CodePlex.
* As a Committer – directly accessing the codebase in the active development repository and directly checking in your changes to the repository.

A contributor is anybody that participates in our forums, reports bugs, and even contributes code to fix minor bugs and other issues. They can propse and submit code revisions and new code by using the **Submit a Patch** feature on CodePlex, [Submit your changes on Codeplex](#_Submit_your_changes). The project team will evaluate submisisons and and integrate them into the main source if they are accepted. For more information on the different levels of participation see the **Contibution Roles** section of the “Overview” document at [http//mbf.codeplex.com/documentation](http://mbf.codeplex.com/documentation).



For details on the different levels of participation see the **Contribution Roles** section in the “Overview” document at <http://mbf.codeplex.com/documentation> and in TFS at ..\Bio\Doc.

## About code contributions

MBF is open to code contributions from the community, with the goal of extending the range of available functionality to researchers and life scientists everywhere. If you need functions that are not in the basic library, you can implement them easily, in a way that works with the existing functions.

We encourage developers who extend the project to contribute their code back to the project as open source, so that the community as a whole can benefit from their work. Microsoft researchers are already using MBF in their research, as are an increasing number of academic and commercial partners. These researchers will be making code contributions to extend the range and power of the project, and we encourage you to do the same.

## Contributions timelines and acceptance considerations

The following are the general guidelines for processing a submittal:

1. You [submit your changes on CodePlex](#_Submit_your_changes).
2. Committers assign Patch adoption to a specific committer within 48hrs, at which point the assigned committer contacts the contributor via CodePlex to let them know they are working on it. They may also request additional information such as unit tests, test results and test data.
3. The assigned committer responds with code review comments within72 hours of submission of all needed materials including documented code, unit tests, test results and test data.
4. If changes are required, the contributor is notified via email and the submittal process is repeated.
5. The contribution is accepted or rejected. Considerations for accepting a contributor's submission include the following:

* Does it meet all posted guidelines posted such as comments, coding, and testing?
* Does the contribution meet the quality bar?
* Were the unit tests included with the submission? Unit tests should be bundled up along with the new code.
* Were all test run and did they all return “green lights?”

1. If the contribution is accepted, the assigned committer merges the contributor's change into the Main development branch within a week of acceptance.

**Note**: If the team is close to a milestone release, then your contribution might be postponed until the next milestone release.

1. If the contribution is rejected, the assigned committer provides the contributor with the reasons for rejection.

## The contribution process

This document describes the process for you to follow when contributing code to the project. The process consists of the following steps:

1. [Contributor getting started](#_Toc288729009)
2. [Review the C# coding and documentation guidelines](#_Toc288729010)
3. [Review existing code](#_Toc288729011)
4. [Create a new CodePlex work item](#_Toc288729012)
5. [Write the new code](#_Toc288729013)
6. [Prepare a file to submit](#_Toc288729014)
7. [Request code review](#_Toc288729015)
8. [Submit your changes on CodePlex](#_Toc288729016)
9. [Respond to code review recommendations](#_Toc288729017)

Note:If you are a professional employee, please check with your employer for your company policy for submitting code to any open source code project**.**

# Contributor getting started

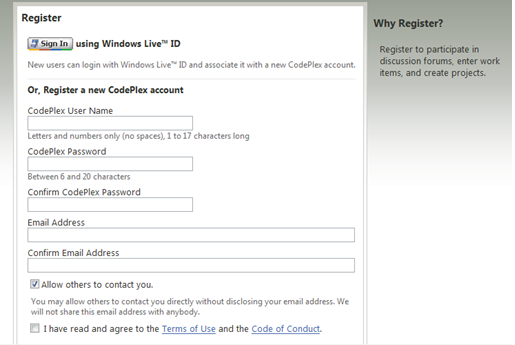
This section describes how to get onboard to participate in the project as a contributor. For more information installing MBF see the “How to Install MBF” section of the “Overview” document on the [Codeplex](http://mbf.codeplex.com/documentation) documentation tab.

## Register at CodePlex and accept Terms of Contribution

You must first sign up for an account on CodePlex. Once registered, you can create projects, enter work items, and participate in discussion forums.

To register on CodePlex

* Follow the instructions at <https://www.codeplex.com/site/register>

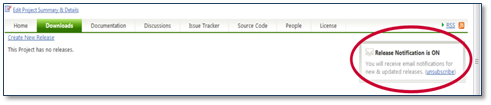


* Once you are registered, your first stop is <http://mbf.codeplex.com>

## Sign up for MBF email and discussion groups

You should ensure that you are signed up to receive email notifications for the following:

* Downloads: Release Notifications
* Discussions: Discussion Notifications
* Issue Tracker: Issue Notifications

  
Setting Release Notifications Choices on CodePlex Projects

Note: You can choose to receive an update on every change or daily summaries, and you can choose RSS feeds instead of email. The important thing is to stay on top of who is doing what on the project, so that you know who you might need to coordinate with.

You can review and manage your email notification settings in your CodePlex profile as well. There might be other CodePlex project notifications you will want to join. This guide focuses on the MBF groups that you should be aware of.

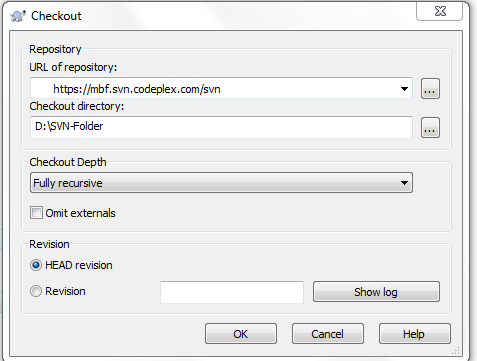
## Install TortoiseSVN

You are required to use TortoiseSVN to prepare your code contribution for submittal using the CodePlex Submit a patch feature. This is the only choice for anonymous access to the MBF project in Codeplex. The url for accessing the CodePlex repository this way is [https://mbf.svn.codeplex.com/svn](https://mbf.svn.codeplex.com/svn/).

TortoiseSVN download is available at [TortoiseSVN.net](http://tortoisesvn.net/downloads.html).

To use TortoiseSVN with CodePlex

1. Download and install TortoiseSVN.
2. Open Windows Explorer and create a new folder to which to download the source code of the CodePlex project.
3. Right click on the new folder and select **SVN Checkout..**.
4. In the **URL of repository** field enter the Subversion URL for the MBF project, <https://mbf.svn.codeplex.com/svn/>.
5. Click **OK** on the SVN **Checkout** screen to download the project source code.



For more information on using TortoiseSVN go to to [CodePlex Information and Discussion](http://codeplex.codeplex.com/wikipage?title=Using%20TortoiseSVN%20with%20CodePlex&referringTitle=Source%20Control).

For more information on CodePlex Submit a patch go [Submit your changes on CodePlex](#_Submit_your_changes).

## Use available resources

Take advantage of the MBF resources in the following table to learn how to use MBF and to follow the community forum.

|  |  |
| --- | --- |
| Resource | Description |
| [Microsoft Biology Initiative](http://research.microsoft.com/bio) – Microsoft Research  <http://research.microsoft.com/bio> | The main MBI website. |
| [Microsoft Biology Foundation - Microsoft Research](http://research.microsoft.com/en-US/projects/bio/mbf.aspx)  [http://research.microsoft.com/bio/mbf.aspx](http://research.microsoft.com/en-us/projects/bio/mbf.aspx) | The main project website. |
| Training  <http://research.microsoft.com/bio/training.aspx> | Training material downloads, including hands-on labs that will help you get started coding. |
| MBI community forum <http://www.getsatisfaction.com/mbi> | A community forum where you can ask and answer questions on the project and receive updates. |

## Dependent software products

The project has the following optional components:

|  |  |
| --- | --- |
| Optional Component | Description |
| Microsoft Silverlight 3 or later [Resources | Microsoft Silverlight](http://www.microsoft.com/silverlight/resources) | Used for the MBF Sequence Assembler application |
| IronPython 2.0.7 Runtime <http://www.codeplex.com/IronPython> | Used for the IronPython scripts |
| Trident Version 1.0 or later <http://tridentworkflow.codeplex.com> | Used for building Trident activities and workflows |
| **Sandcastle Help File Builder** <http://shfb.codeplex.com/>/ | Used to automatically generate a help file for the APIs |
| **FxCop** <http://www.microsoft.com/downloads/en/details.aspx?displaylang=en&FamilyID=917023f6-d5b7-41bb-bbc0-411a7d66cf3c> | To check for possible design, localization, performance, and security improvements in .NET managed assemblies |

**Note**: To compile the project, Microsoft Visual Studio® 2010 or later is required.

For academic researchers. You can receive free Microsoft software to assist in working with the project. To download and install the latest version of Microsoft Visual Studio (VS2010), visit the Microsoft [DreamSpark](https://www.dreamspark.com/) web site at <https://www.dreamspark.com>.

You will also find other optional components on DreamSpark that you might like to use in your work.

# Review the C# coding and documentation guidelines

When developing code for the project, follow the guidelines in the following documents on Codeplex, <http://mbf.codeplex.com/documentation>:

|  |  |
| --- | --- |
| Document | Description |
| **C# Coding Standards** | Presents guidelines for developing code for the project. One of the key components to delivering on the MBF promise of higher productivity is the ability to provide a consistent approach to the programming model and stylistic conventions used throughout MBF development. |
| **How to Write Code Comments in Source Code** | Discusses how to write good documentation comments, specifically for projects that will produce an API reference directly from code comments. The quality bar is high for such projects, because there is little opportunity to modify the text after it is harvested. |
| **Testing\_Guide** | Discusses how to create and run the tests provided with the project and any new tests for your code contribution, either as unit test or automation. |
| **Contributors Template** | Discusses how to write good documentation comments, specifically for projects that will produce an API reference directly from code comments. The quality bar is high for such projects, because there is little opportunity to modify the text after it is harvested. |

# Review existing code

As participant in an open source project, you can evaluate the existing code first hand. Although most of the detailed technical information is available in the SDK documentation, you might find it easier to simply examine the code files and associated comments. If you use code editing software with reference/use tools such as Visual Studio, it is also helpful to traverse through the code via reference.

One other technique that we highly recommend is to build the samples provided and run them under the debugger. By setting breakpoints and stepping through actively running code, you will be able to see the order of operation (at least for a particular sample). Also, by observing the data register during the debugging session, you can follow along with how the data is represented and modified throughout the sequence of execution.

Finally, by taking a look at what others have provided as code changes to the base framework, you can get an idea of what type of code is expected in a contribution. Changelists and ShelveSets can be quite interesting to evaluate, and will provide context for both the new and old code and the related differences.

# Create a new CodePlex work item

The MBF community uses the CodePlex **Issue Tracker** to create and track issues such as feature work, product issues, and tasks.

For general help on the Issue Tracker, see [the CodePlex Information and Discussion wiki](http://codeplex.codeplex.com/wikipage?title=Issue%20Tracker&referringTitle=CodePlex%20Documentation).

To create a work item on Issue Tracker

1. Sign in to CodePlex and go to the MBF project.

2. Click Issue Tracker.

3. Click Create New Item.



4. Set Status to Proposed.

5. Set Type to Task.

6. Enter a clear title and description for your work item.

7. Fill out the remainder of the form as appropriate and click Save.

# Write the new code

At this point you have done everything necessary to begin writing and contributing code as a committer. When writing your new code, following the coding and commenting guidelines described in the [Review the C# coding and documentation guidelines](#_Review_the_C#) section. Install and build the code as described in the “Overview.docx document” on Codeplex, <http://mbf.codeplex.com/documentation>. Then procede with the following steps:

To write new code

1. Write your new code.
2. [Write documentation for new code](#_Write_documentation_for).
3. [Write and run unit tests for the new code](#_Write_unit_tests).
4. [Run the tests included with the installed project and confirm all tests pass](#_Run_tests).
5. [Prepare a file to submit](#_Prepare__a) via CodePlex.
6. [Request code review](#_Request_code_review).
7. [Submit your changes on CodePlex](#_Submit_your_changes).

## Write documentation for the code

If your code includes new features for users, then fill out the “Contributors Template”, found at <http://mbf.codeplex.com/documentation>.

If your new code includes new APIs for developers, then be sure to:

* Comment the APIs by following the guidelines in “How to Write Code Comments in Source Code” document found at <http://mbf.codeplex.com/documentation>.
* The Sandcastle project is run as part of the build. Building the project will automatically produce the reference chm. For more information on using Sandcastle Help File Builder go to <http://shfb.codeplex.com/>.

**Important**: Always run **FxCop** as described in [Run FxCop](#_Run_FxCop).

## Write unit tests for the new code

MBI uses VSTest as the primary means for ensuring code quality, which makes it easy to develop, integrate, and run test code against your contributions.

You must create a new test for your code contribution, either as unit test for small changes or automation tests for new features and large changes, as described in the “Testing\_Guide.docx” document at <http://mbf.codeplex.com/documentation>.

For more information on using Visual Studio 2010 test features see [Testing the Application](http://msdn.microsoft.com/en-us/library/ms182409.aspx) on MSDN.

## Run tests

The final step in producing new code is to run the tests, as described in the “Testing Guide” document. All test cases must pass before you [prepare a file to submit](#_Prepare__a) and submit your [changes on CodePlex](#_Submit_your_changes).

To test new code

1. Run the new unit tests you have created and any new automation suite tests.
2. Run the tests provided with the installed project.
3. If any tests fail, fix your code and try them again.

## Run FxCop

Run **FxCop** when you finish coding. Correct any errors it finds. **FxCop** analyzes programming elements in managed assemblies by using rules that return informational messages when the rules are violated. Messages identify any relevant programming and design issues and, when it is possible, supply information about how to fix the issues.

**Note** Some errors reported by FxCop may not be consistent with the project Coding Guidelines. In general, FxCop issues should be addressed, but not if they conflict with the project guidelines. Use good judgment in applying FxCop recommendations, and it is acceptable to request new exceptions to the FxCop errors as well. The important thing to ensure is that the code you write compiles cleanly with no errors or warnings.

# Prepare a file to submit

When you are satisified with your code and all tests have passed prepare your code for submission.

For open source projects the change is managed by submitting a patch file to the development team, who do have write access. They review the patch, and then either submit it to the repository or reject it back to the author.

Patch files are simply Unified-Diff files showing the differences between your working copy and the last update from the repository, which in this case is the downloaded version from CodePlex.

To create a patch

1. Use the *TortoiseSVN → Create Patch* command on the parent folder.

**Note** This is the only choice to access to the project in Codeplex. The url for accessing our repository this way is [https://mbf.svn.codeplex.com/svn](https://mbf.svn.codeplex.com/svn/).

1. In the **Create Patch** dialog box select the files to include.
   1. Make sure you include all unit tests.
2. Click **OK** and this will produce a single file containing a summary of all the changes you have made to the selected files since the last update from the repository.
3. Submit the file, see the [Submit your changes on CodePlex](#_Submit_your_changes) section.

For more information go to [Creating and Applying Patches](http://tortoisesvn.net/docs/release/TortoiseSVN_en/tsvn-dug-patch.html) using TortoiseSVN on Tortoise.net.

# Request code review

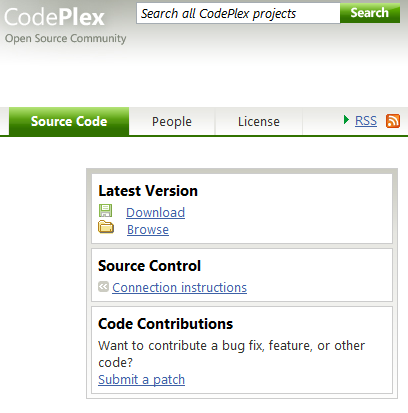
It is always a good practice to have your code reviewed by your team or another developer before submitting it on CodePlex.

# Submit your changes on CodePlex

You can submit your new code on CodePlex using **Submit Patches**.

To submit your changes

* Click Submit a patch on the [MBF CodePlex Source Code](http://mbf.codeplex.com/SourceControl/list/changesets) page.



Project coordinators will review your submission. The time for review varies, depending upon where the project team is in its schedule. If the team is close to a milestone release, then your contribution might be postponed until the next milestone release. The general guideline for processing a submittal is:

1. Contributor submits a Patch via CodePlex.
2. Committers get automatic notification via email.
3. Committers assign Patch adoption to a specific committer within 48hrs, at which point the assigned committer contacts the contributor via CodePlex to let them know we are working on it. They may also request additional information such as unit tests, test results and test data.
4. The assigned committer responds with code review comments within72 hours of submission of all needed materials including documented code, unit tests, test results and test data.
5. If changes are required, the contributor is notified via email and the submittal process is repeated.
6. The contribution is accepted or rejected.
   1. If the contribution is accepted, the assigned committer merges the contributor's change into the Main development branch within a week of acceptance.
   2. If the contribution is rejected, the assigned committer contacts the contributor with denial reasons.

**Note**: Always include all tests for your new code with your submission.

# Respond to code review recommendations

Project coordinators may respond to your CodePlex submission with recommendations for your contribution. Implement any recommendations from the project coordinators and repeat the cycle – test, have your code reviewed, submit a new shelveset on CodePlex.