Getting Started with ARM® mbed™

Introduction

The purpose of this workshop is to help users get acquainted with the ARM® mbed™ tools. The entry point for the tools is http://developer.mbed.org. There you will find hardware platforms, component libraries and a web-hosted IDE / compiler. This means that the mbed tools work on all operating systems (Linux, Mac and Windows). The tools are free for developers and the software is available under the commercially friendly Apache 2.0 license.

The tools consist of an online components database with drivers and example code, an online compiler for importing, compiling and sharing projects, and boards that are very simple to program using a drag and drop interface.

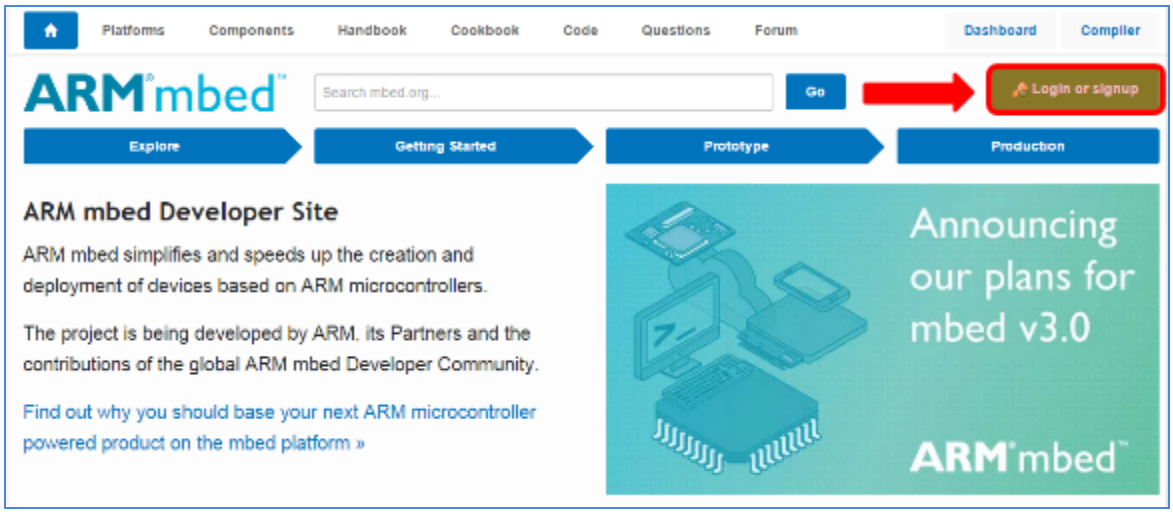
Please note that there is also a YouTube playlist for getting started with mbed. This document is best viewed as a digital document. Here is a permalink to the document in case you are viewing it offline.

https://docs.google.com/document/d/1E5bq7uKYGjcr3K5aYHLN8D3uoakf6ARhNZF11YpkcAE/

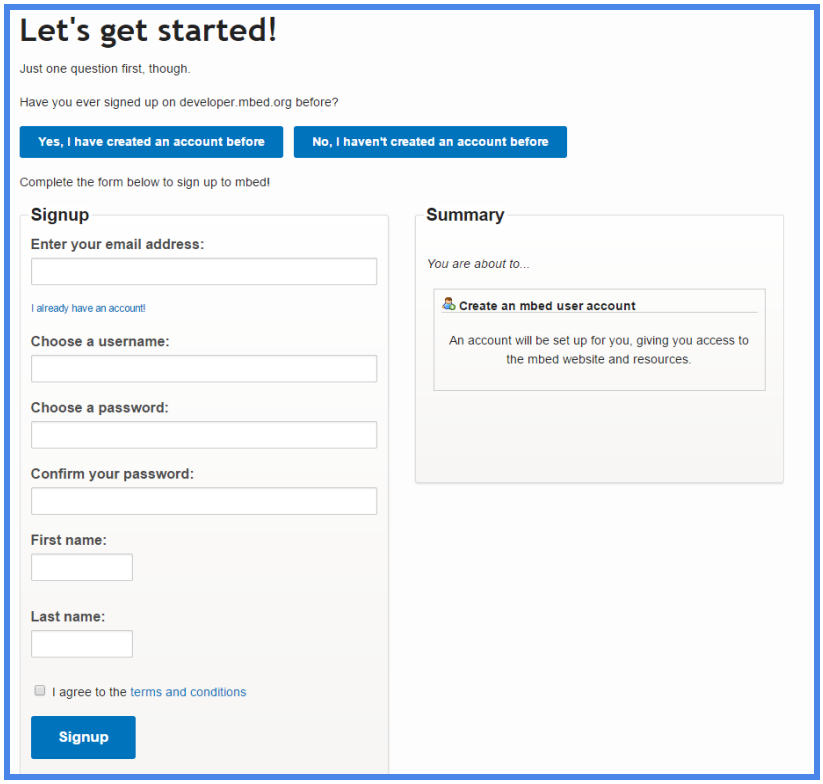
Table of Contents:

Creating an mbed Account

1) Go to http://developer.mbed.org and sign up for an account. Alternatively, you can open the mbed.html that is on every mbed development board.



2) Fill in your information. The username you choose here will be what the community sees on your code and all comments you make.

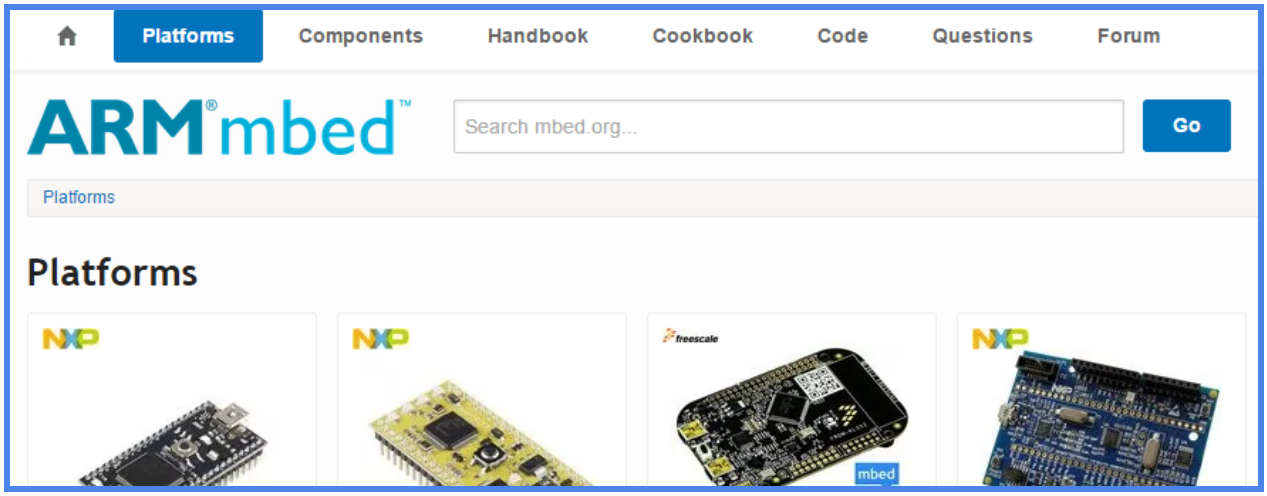


The Online Community

The mbed tools include an online compiler, loads of ready-to-use component code, example programs and much more. Official APIs can be found on the Handbook page, community-contributed high quality code projects can be found on the Cookbook page, and the Forums are an asset for getting answers to a wide range of questions efficiently

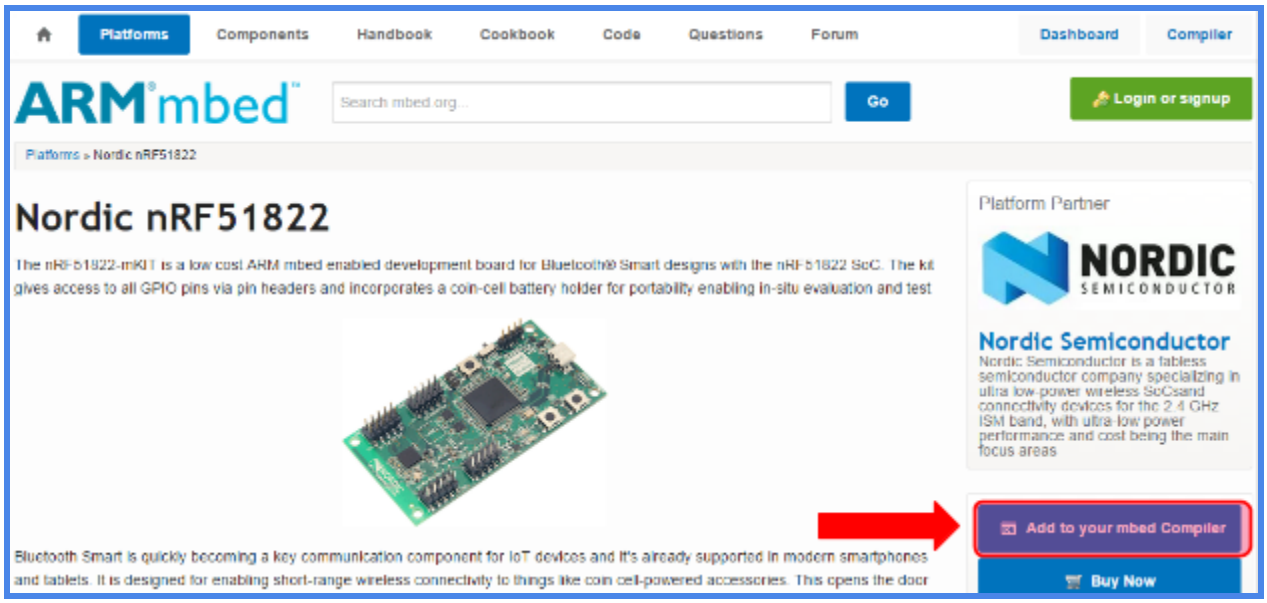
Platforms Page

The Platforms page contains a list of every platform officially supported in the mbed ecosystem. There are links on the platform pages for each board to add the board to your compiler, to buy the board, pinout diagrams, firmware updates and bug fixes. It is important that you look at the platform page for the board you are using to find what the pin names are called and check for firmware updates..



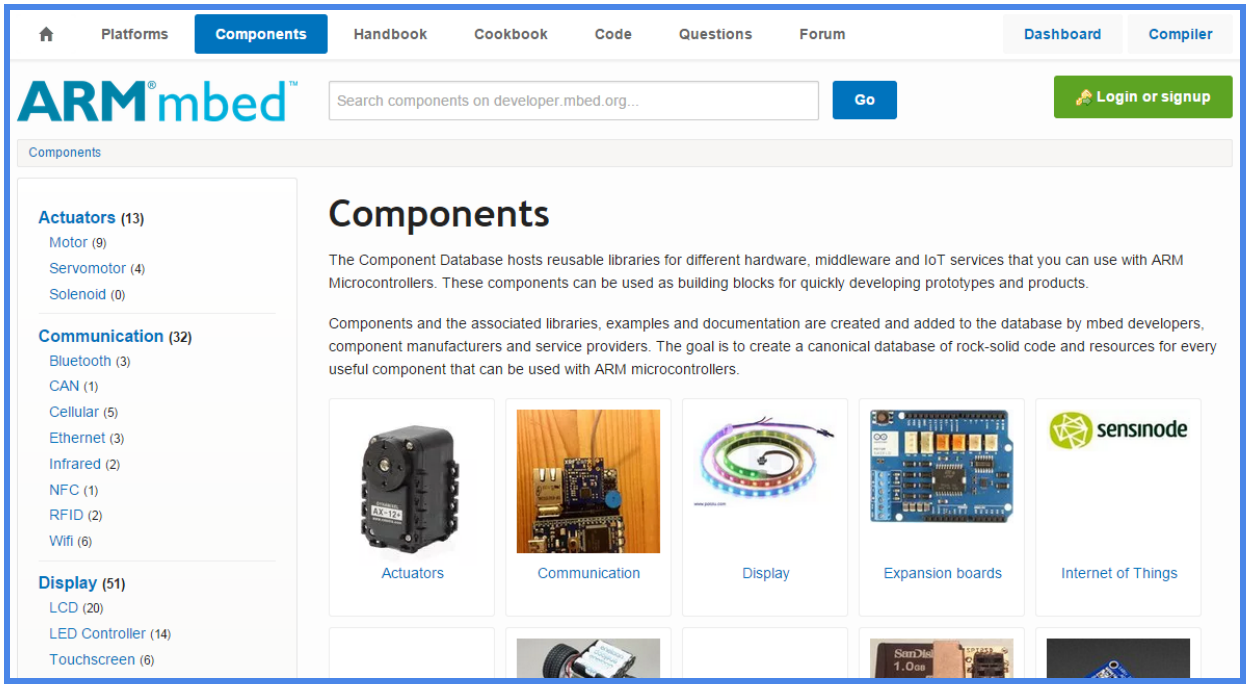
If you want to compile code for a board you should go to its platform page and click the **Add to Compiler** button. That's it! Now when you are in the compiler you can compile code for the board you selected.

Tip: Make sure you are logged in to do this.



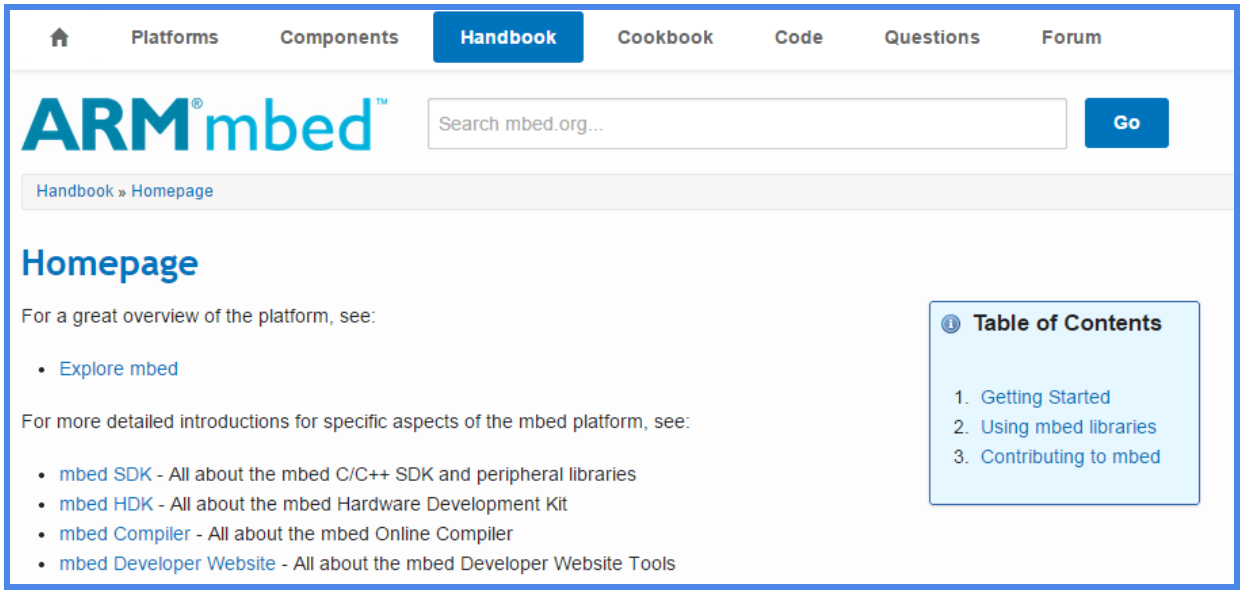
Components Database

The components database consists of libraries with examples and schematics for a wide variety of hardware components. The goal of this section is to eliminate the frustration of using a new part by leveraging pre-written drivers and sample applications for a given component. For example, say you want an RFID transceiver; you would search the components database for “RFID” and pick the appropriate transceiver. Then you would have the option of importing sample code into your compiler to take the component for a test drive.



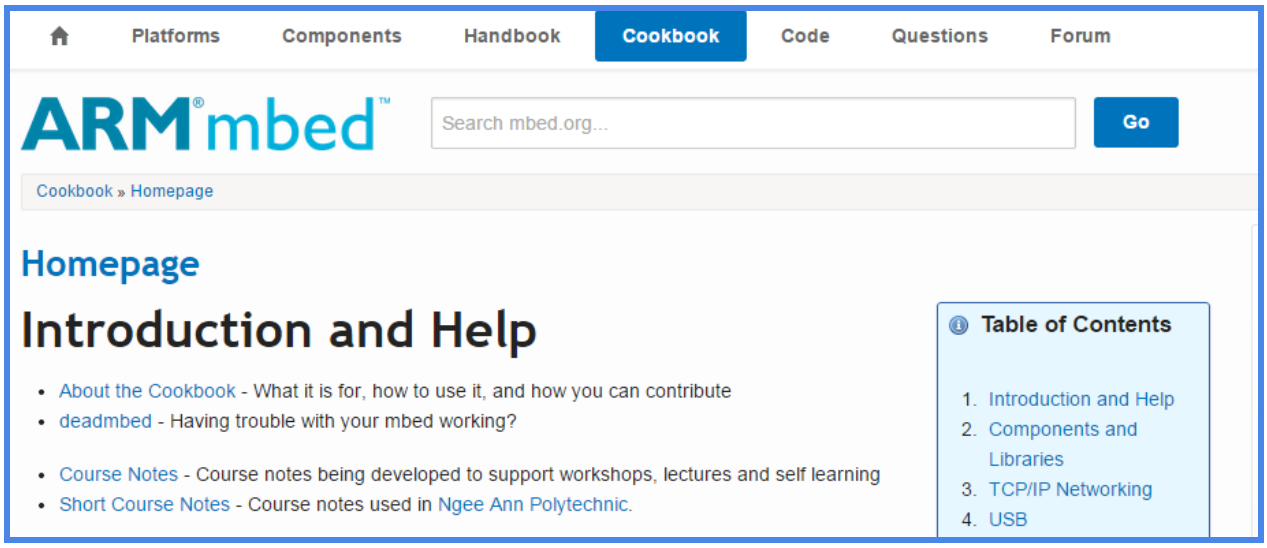
Handbook APIs

The [Handbook page]documents the official mbed software interface. The handbook is where you would find the APIs for standard microcontroller software interfaces such as SPI, ADC and UART.



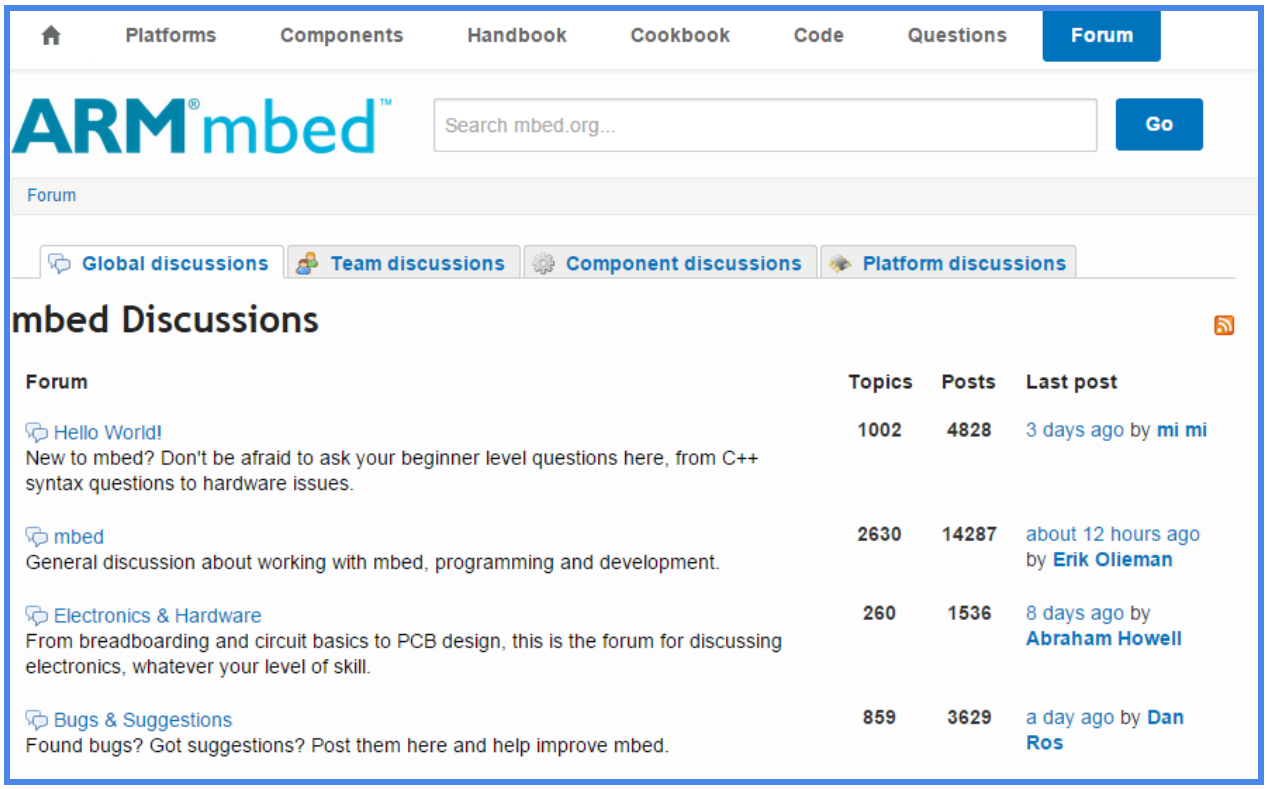
Cookbook Examples

The Cookbook page is a community-controlled page where anyone can make contributions and show off projects. The cookbook is in the process of being phased out and replaced by the components database as a place to share drivers, as well as by the Completed Projects page as a place to show off projects. While you can still find some solid community contributed content on the Cookbook page it is not recommended to put new content there.



Forums and Questions

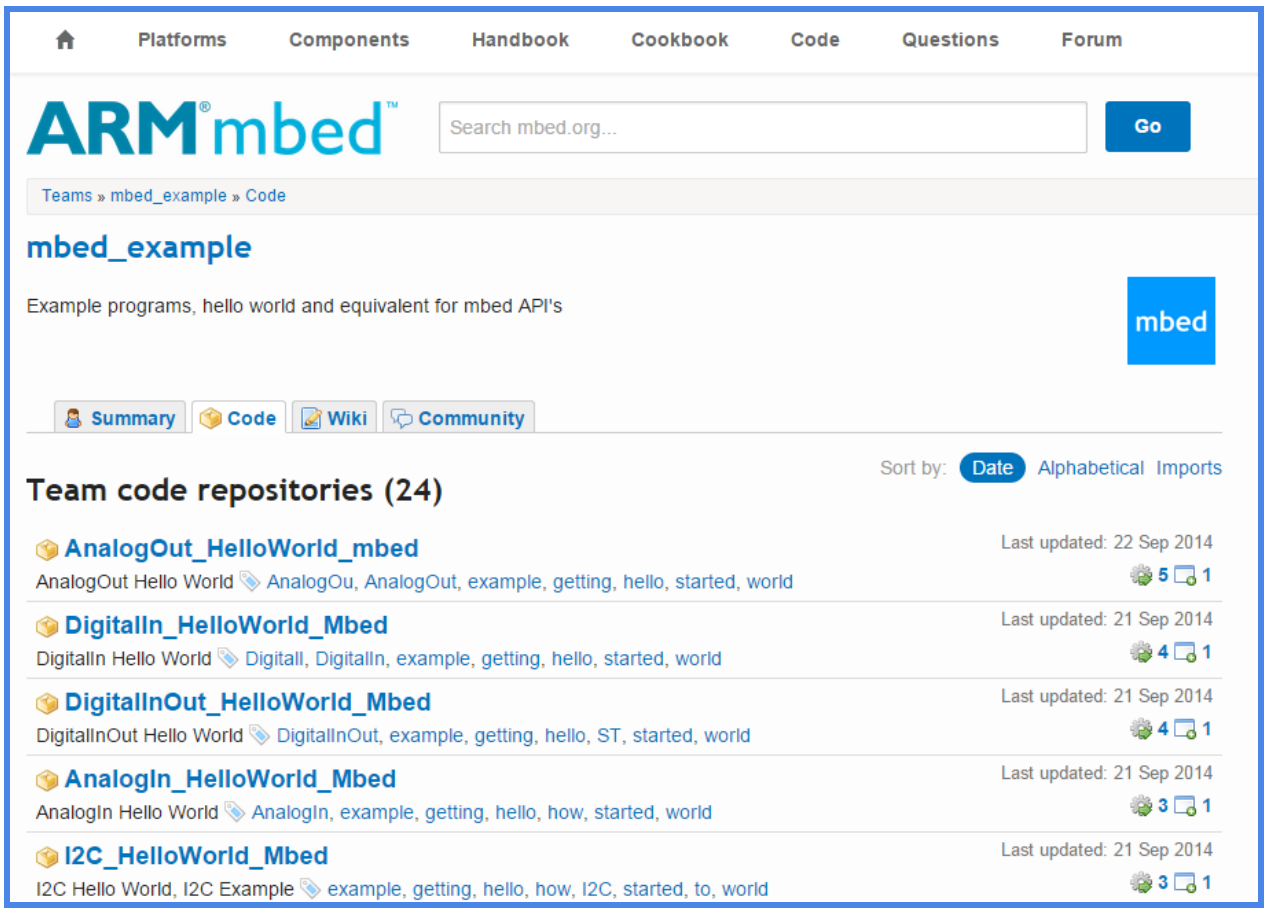
The forum is an excellent place to ask questions about anything and get a quick response from the community. It is mostly moderated by community members and is a source for tips, tricks and advice for the mbed platforms.



Every code example / wiki page has a questions section at the bottom of the page. All of the questions from the website are also aggregated in chronological order under the Questions tab.

Code Examples

You can find code examples for existing components in the components database or by using the search function. The mbed_examples team page has great sample projects demonstrating the mbed API.



The Online Compiler

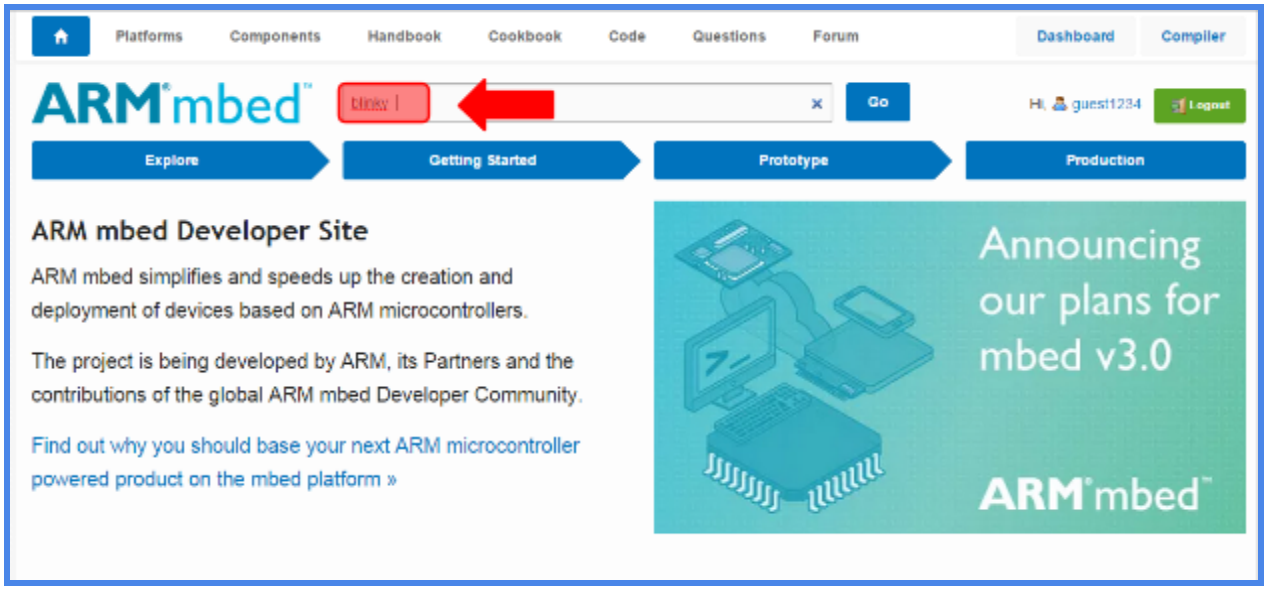
The online compiler enables you to either write your code from scratch or import an existing project and modify it to suit your needs. The compiler allows for customization of platform target, drivers, multiple projects and more.

Import Code to Online Compiler

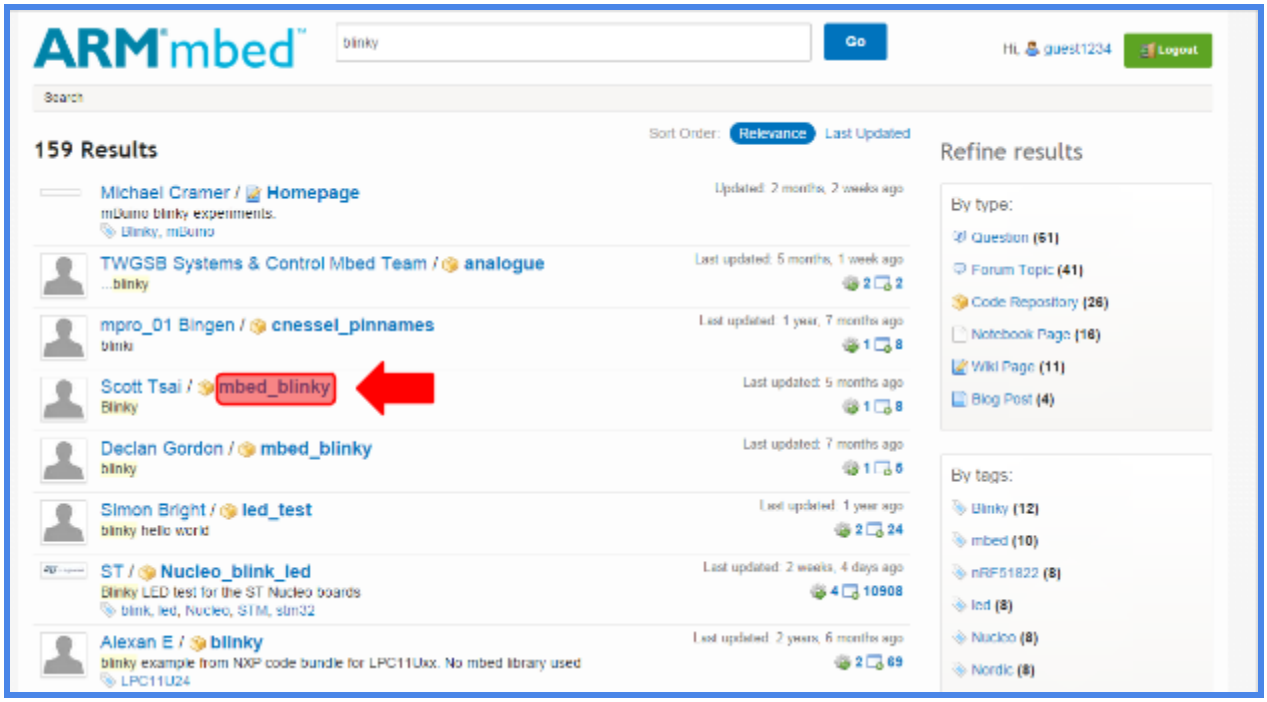
There are two methods of importing the code into the online compiler: using the site's search bar and using the compiler's Import button.

The Search Bar Method

The search bar method starts at the top of the page on developer.mbed.org. Input the keywords to whatever project you are interested in finding:



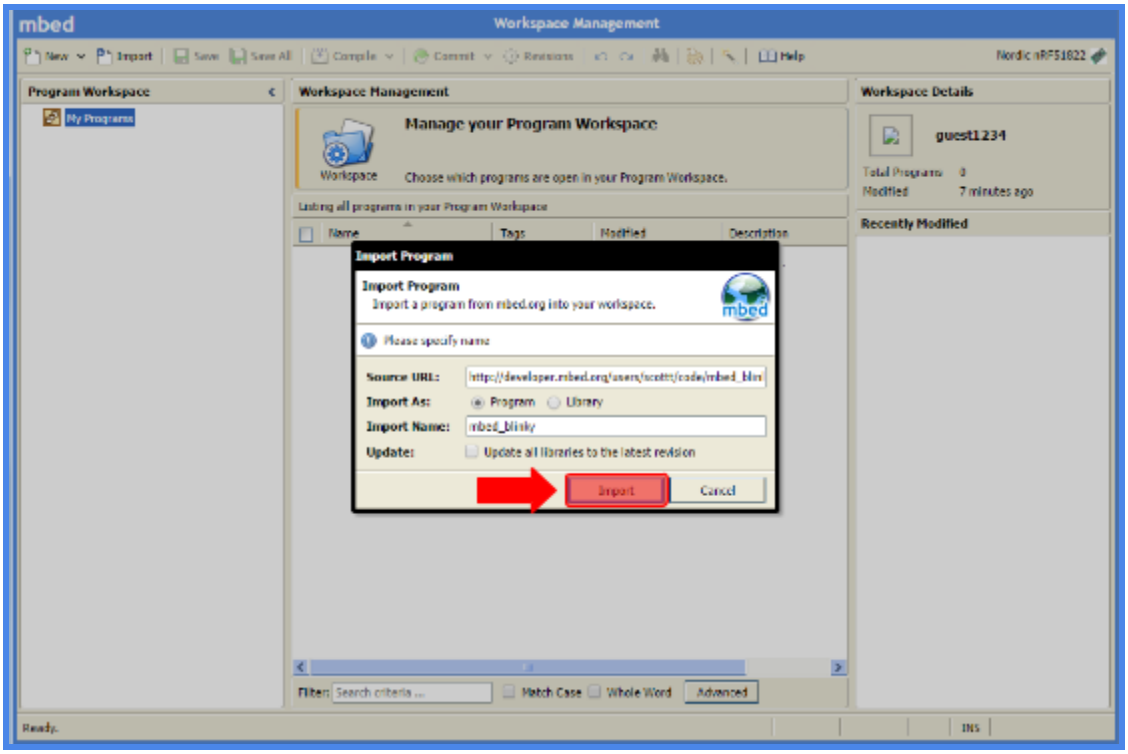
The search results are sorted by relevancy; you can search by update date (newest first) by clicking **Last Updated**. Click on a result link if you want to take a closer look at any particular project or import it.



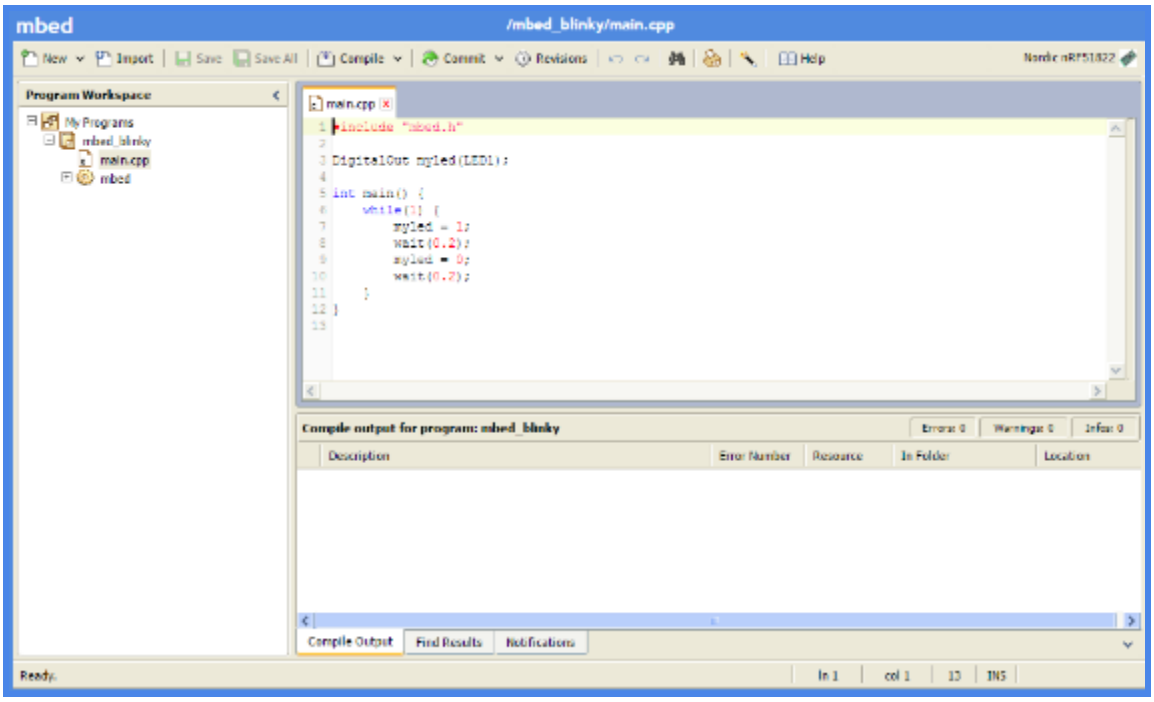
If you decide to use a project:

1. Click the **Import this program** button on the right-hand side of the project's page to add it to your compiler.

2. If you don’t already have a compiler running, it will open in a new tab. The compiler will ask you to confirm the import. Click **Import** to complete the action.



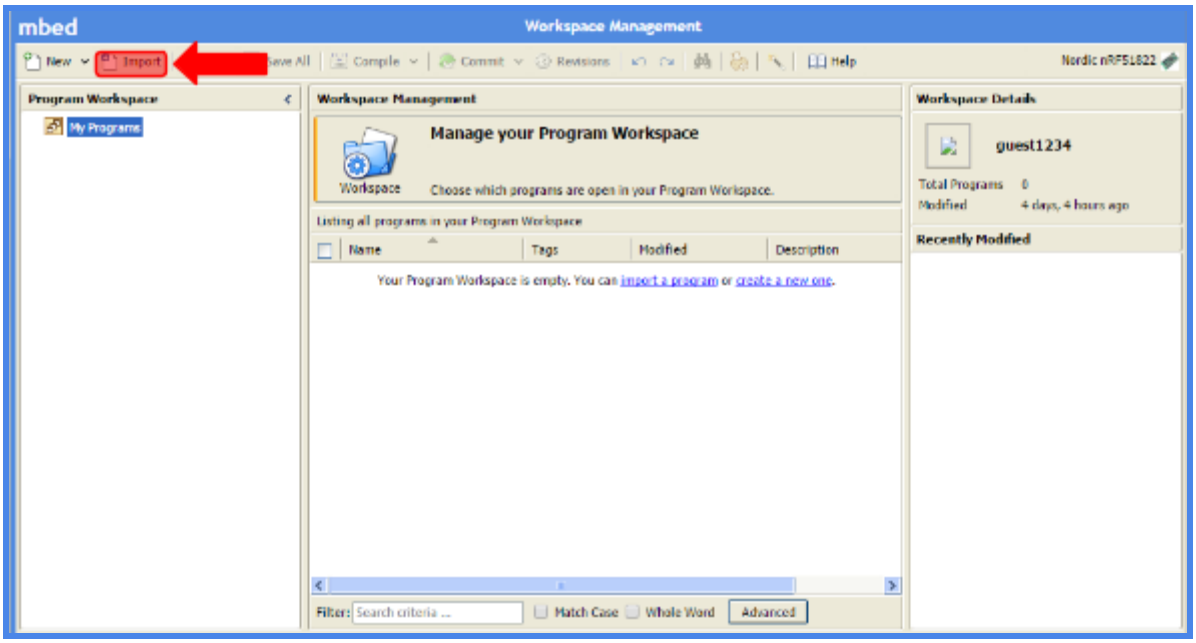
3. You can now edit the files as you see fit. To edit a file click on it in the left hand view and it will load in the central view.



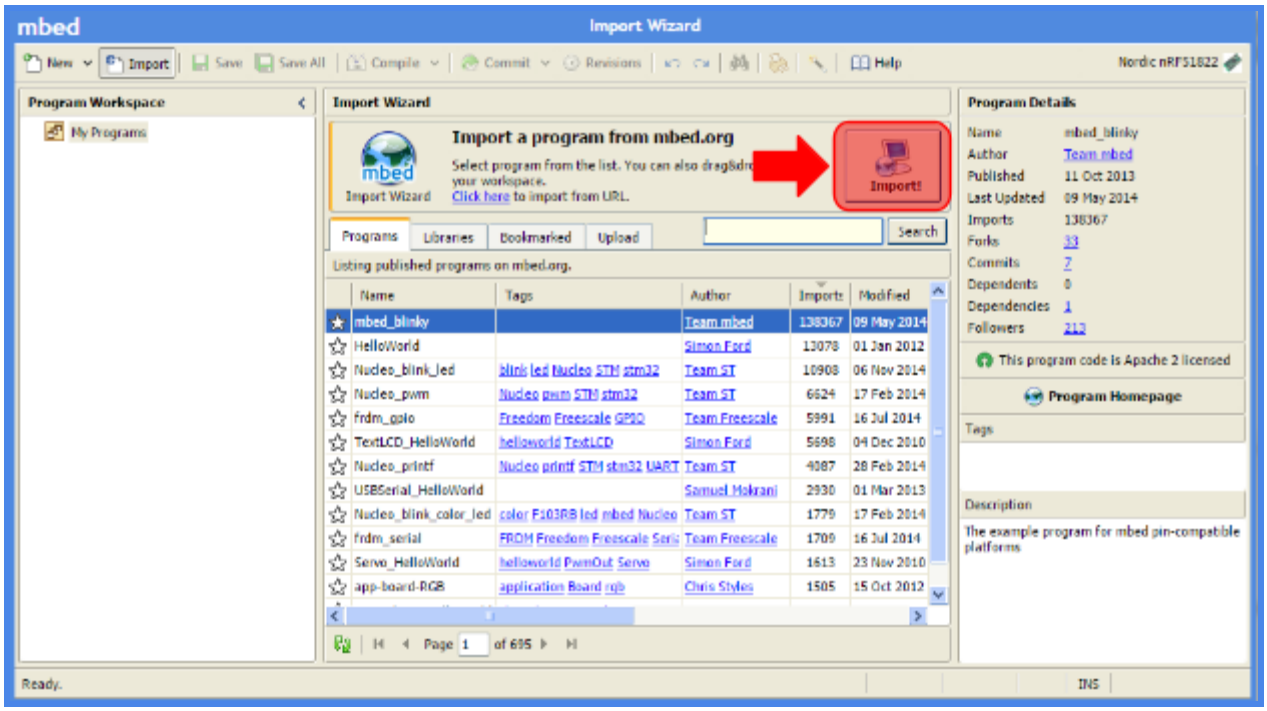
Compiler Import Method

The other option is to import programs directly from the online compiler workspace.

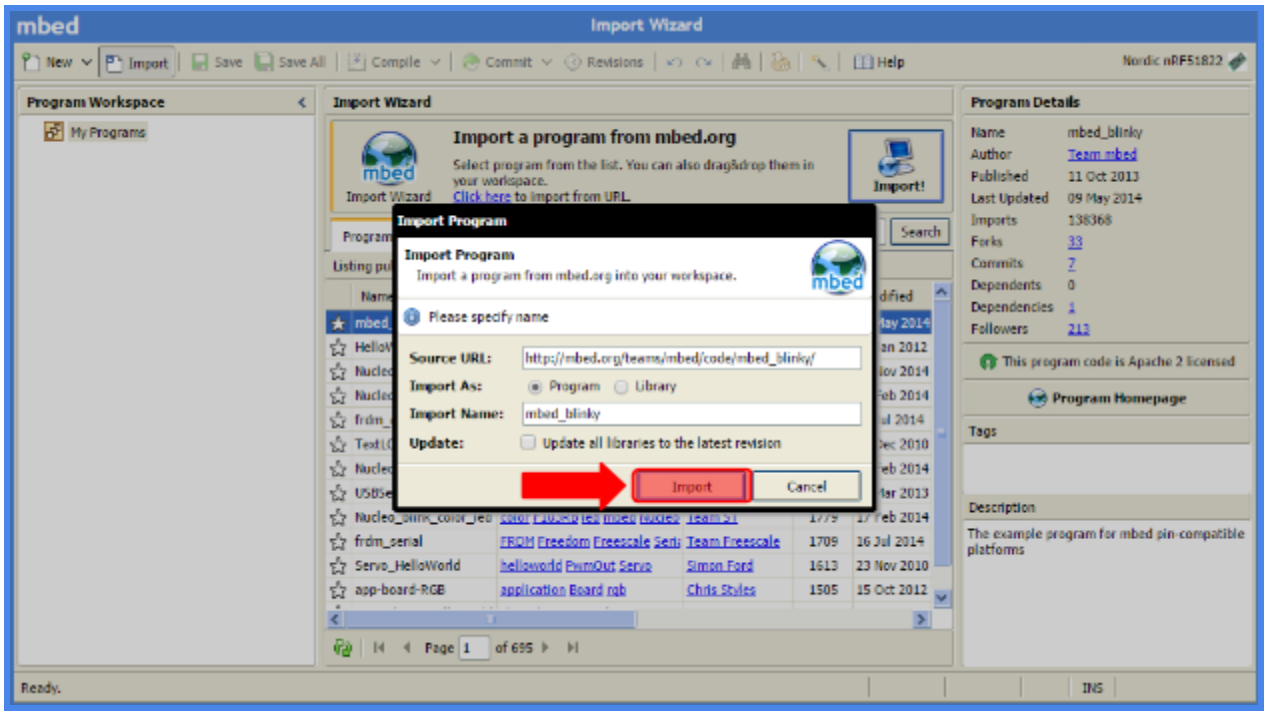
1. Click the **Import** button on the top-left corner of the compiler.



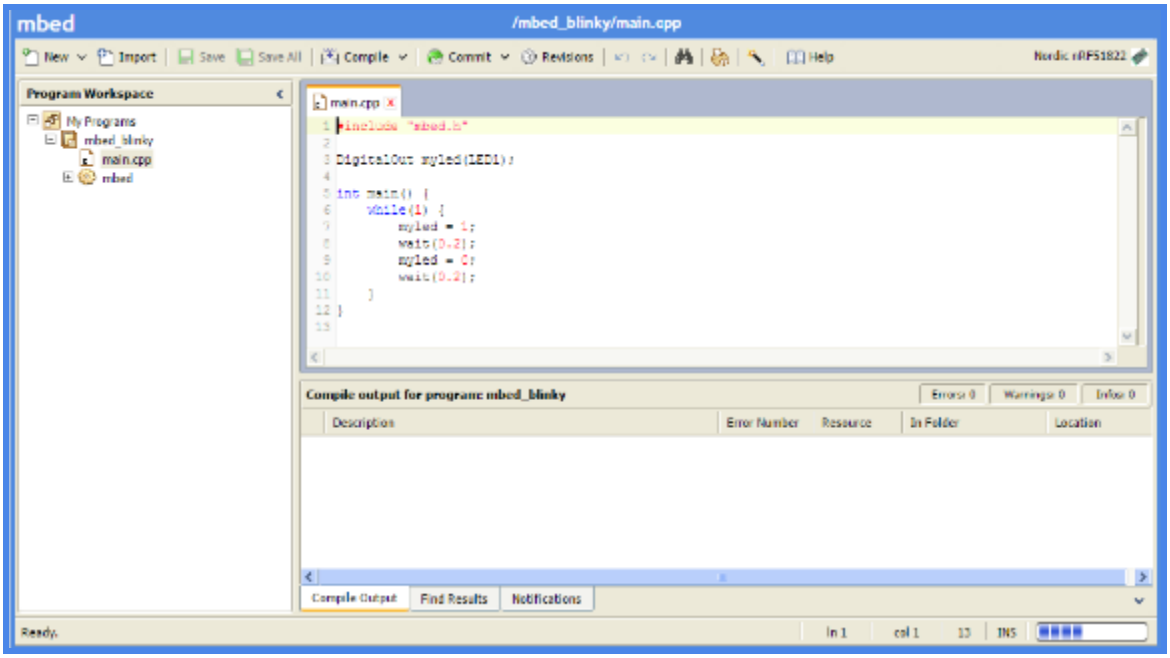
2. The **Import Wizard** opens, with a list of the most popular project downloads. The search bar in the top right corner can be used to find a different project.
3. When you've found the project you want, highlight it and click **Import**.



4. The compiler will ask you to confirm the import. Click **Import** to complete the action.



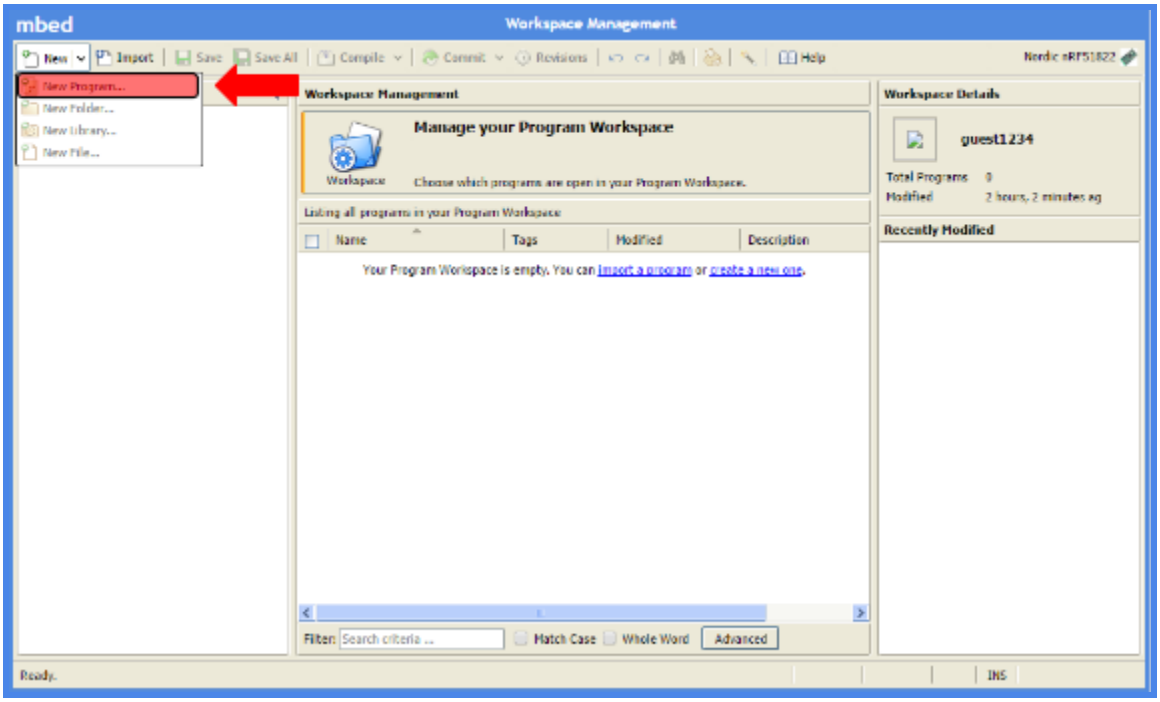
5. You can now edit the files as you see fit. To edit a file click on it in the left hand view and it will load in the central view.



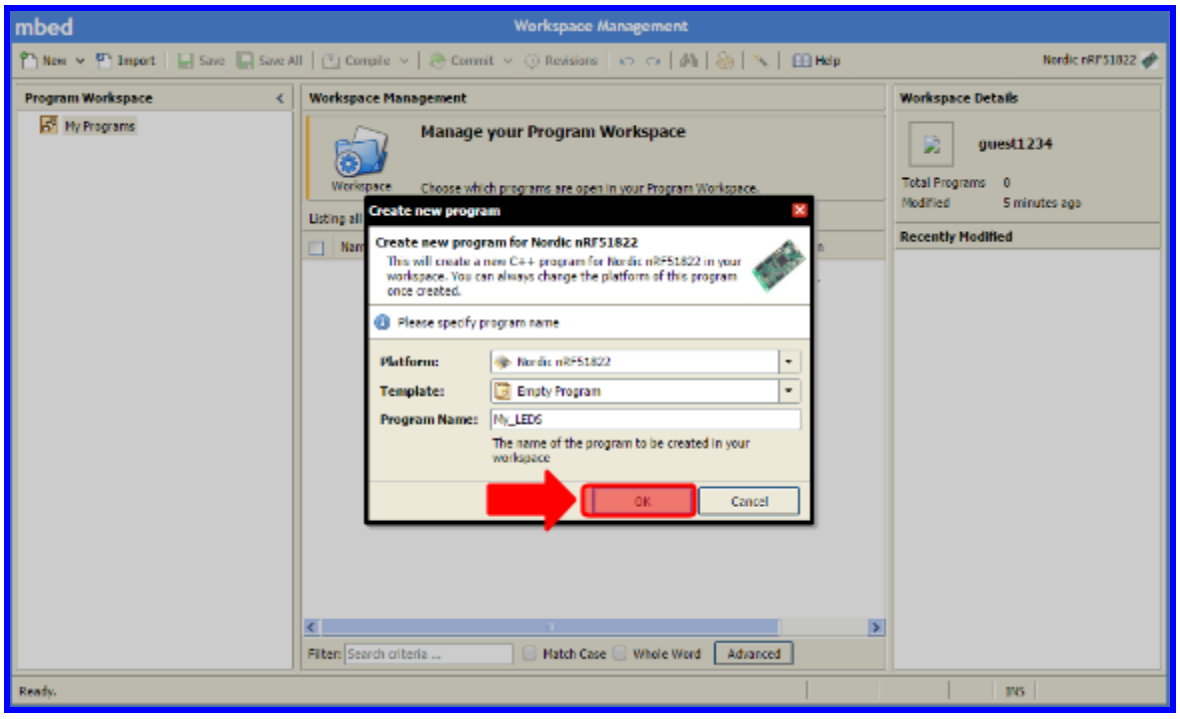
Make a Project From Scratch

To make a project from scratch:

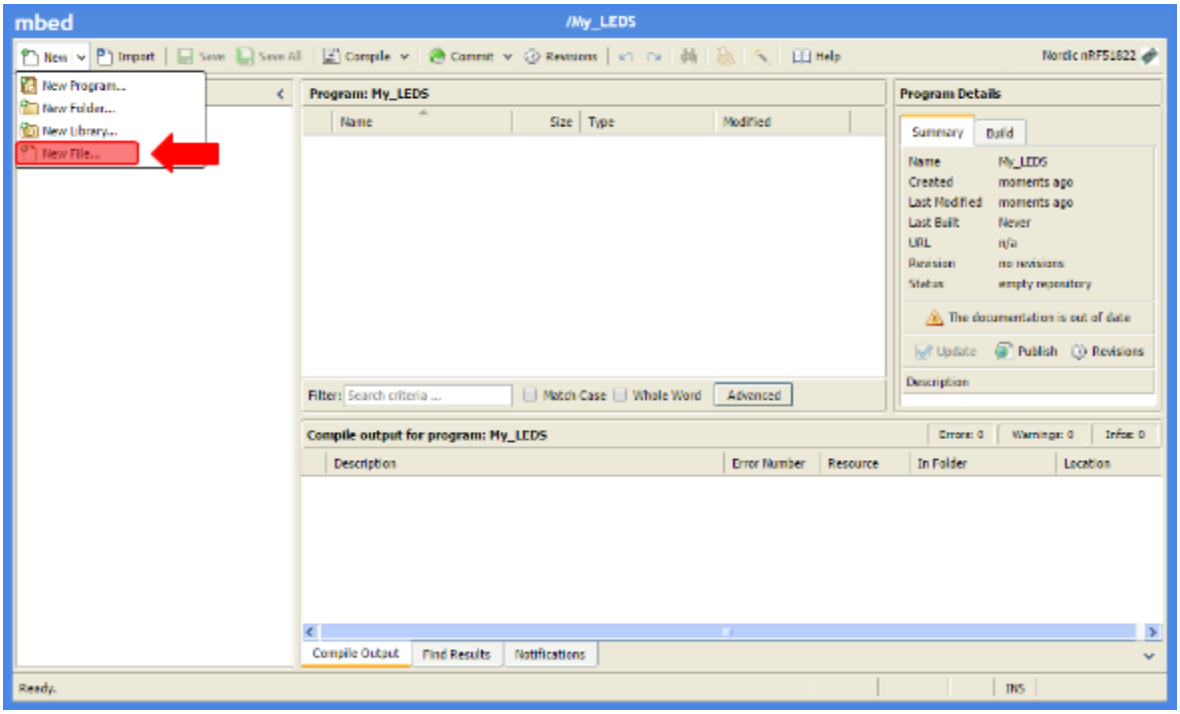
1. Click **New>New Program.**



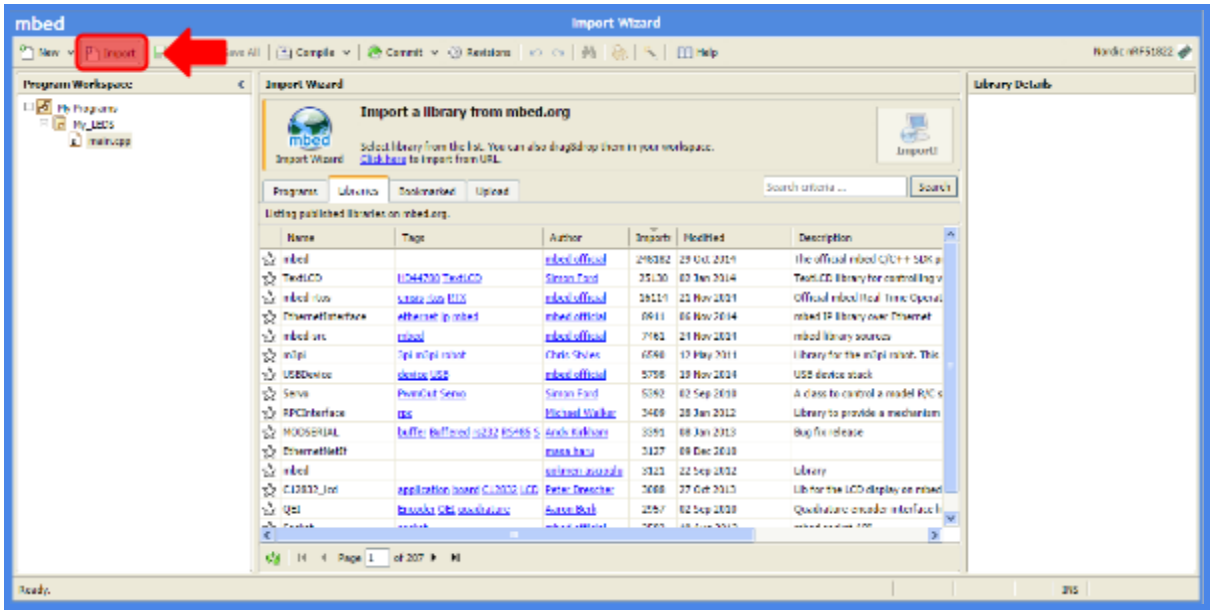
2. Select your platform.
3. You can work from a template, or create an entirely new program.
4. Name your program.
5. Click **OK**.



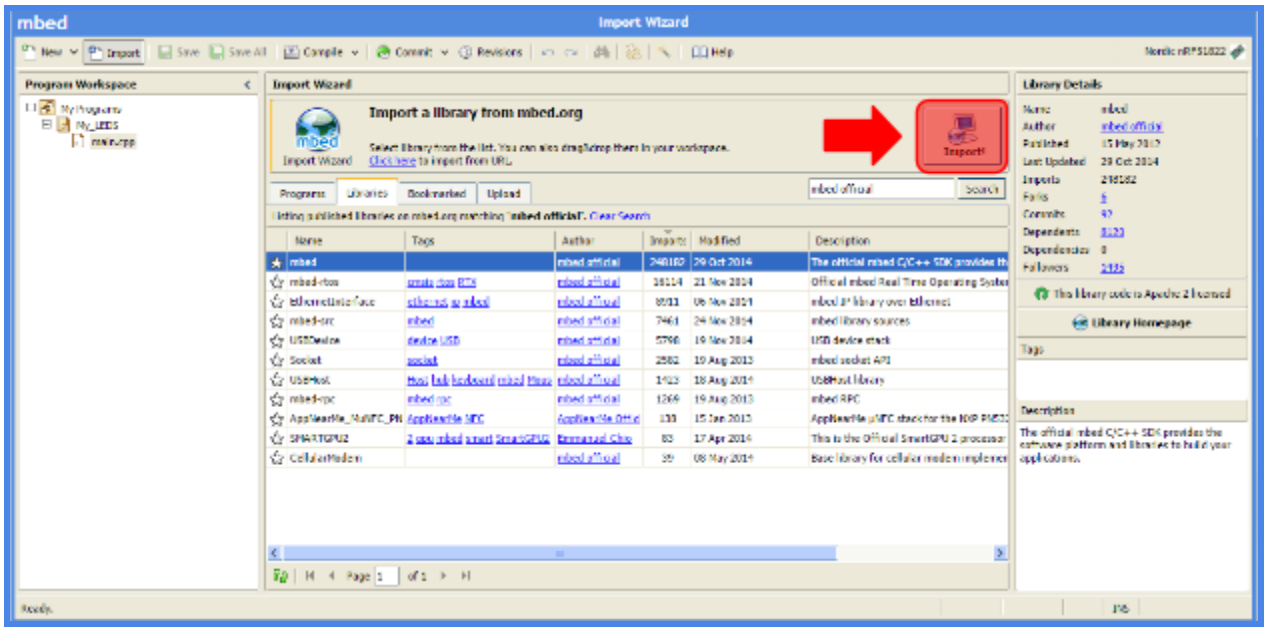
6. Create a main.cpp file to the project. Click **New>New File**.



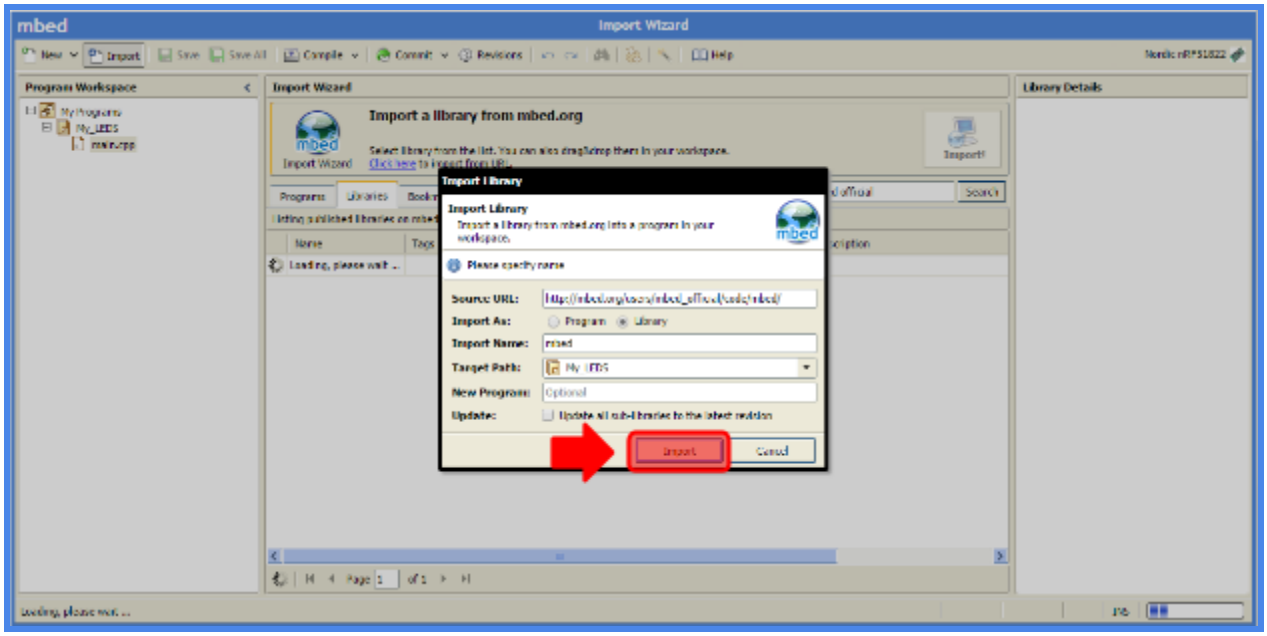
7. You can create your own libraries by clicking New>New File.
8. Alternatively, you can import the libraries that you need from the available list, using the same procedure you used to import code.;
    a. Click Import.
    b. In the Import Wizard, click the Libraries tab to view a list of libraries. You can also search the libraries by using the Search bar.



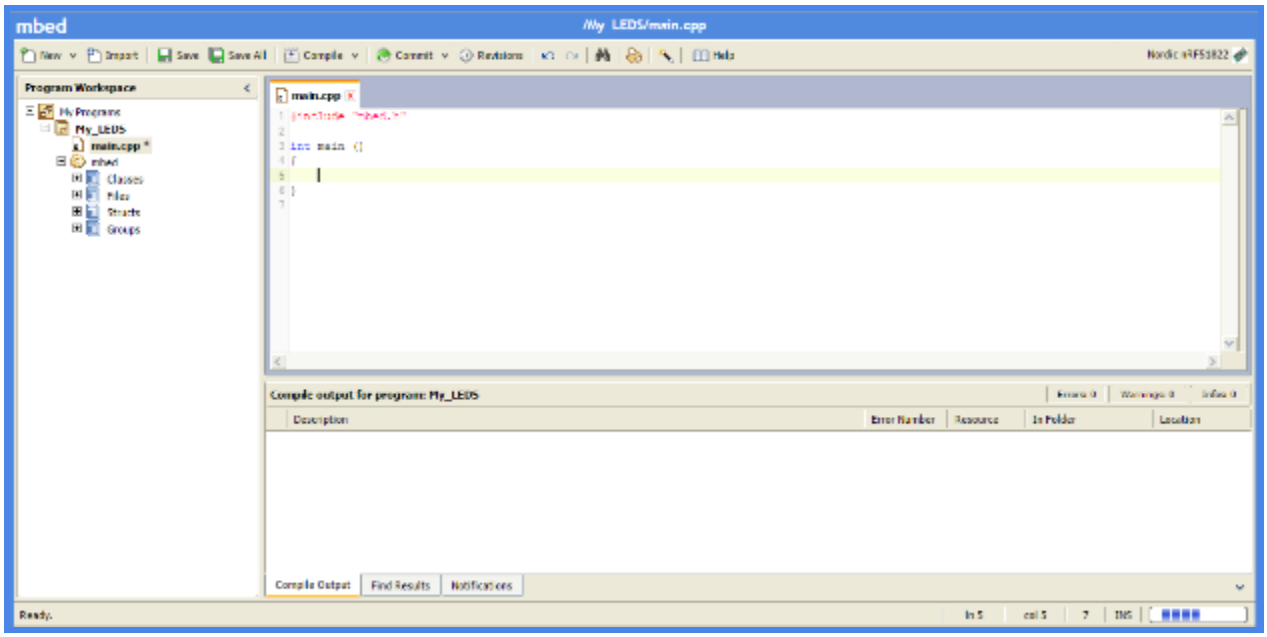
    c. Highlight the library you need and click **Import** in the wizard.



d. The compiler will ask you to confirm the import. Click **Import** to complete the action.

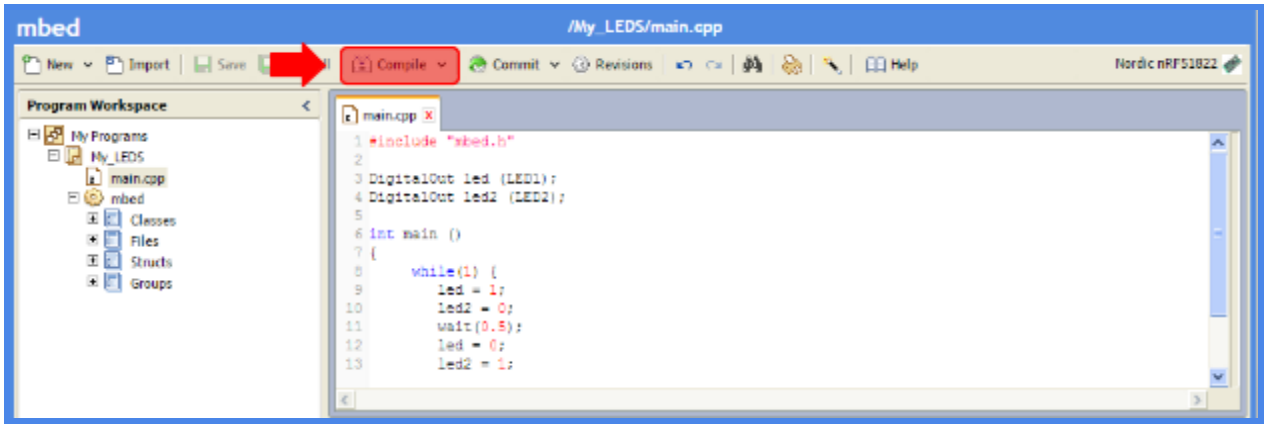


9. Code the features you need, save and compile when you are ready to test. See below for compilation instructions.


mbed
/My_LEDS/main.cpp
New
Import
Save
Save All
Compile
Commit
Revisions
Help
Nordic nRF51822
Program Workspace
My Programs
My_LEDS
main.cpp *
mbed
Classes
Files
Structs
Groups
main.cpp
#include "mbed.h"
int main () {
}
Compile output for program: My_LEDS
Description
Error Number
Resource
In Folder
Location
Compile Output
Find Results
Notifications
Ready.
ln 5
col 5
7
INS

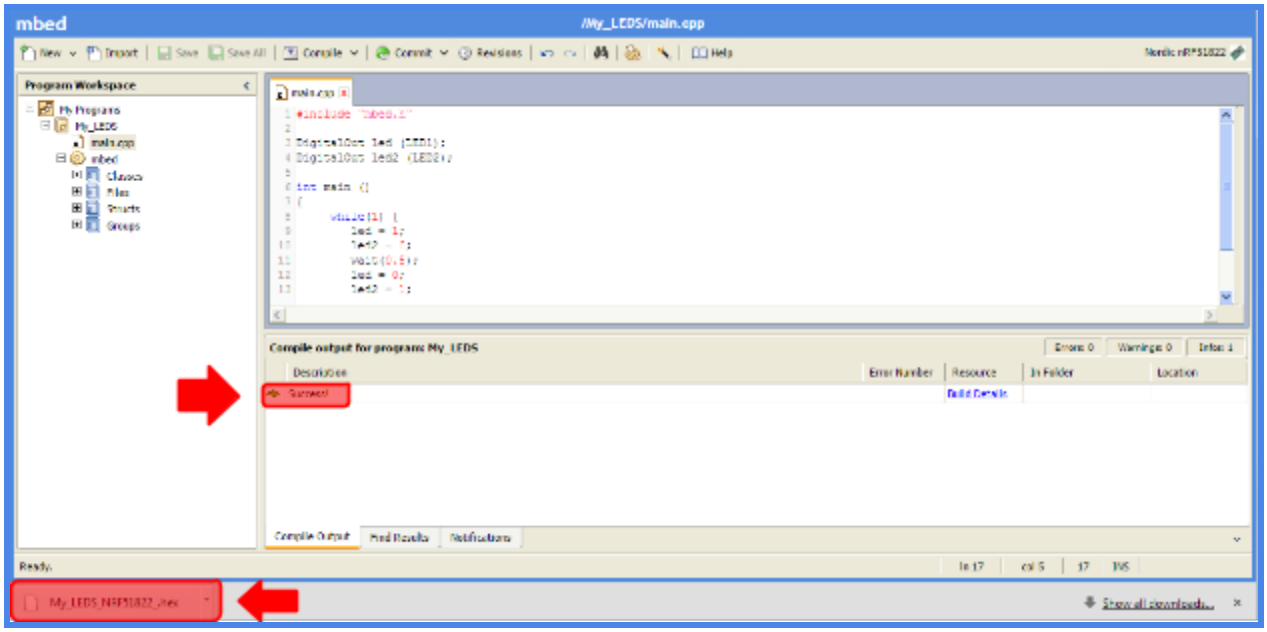
Compiling Code Online

Once you are ready to test your project click **Compile** at the top of the page.



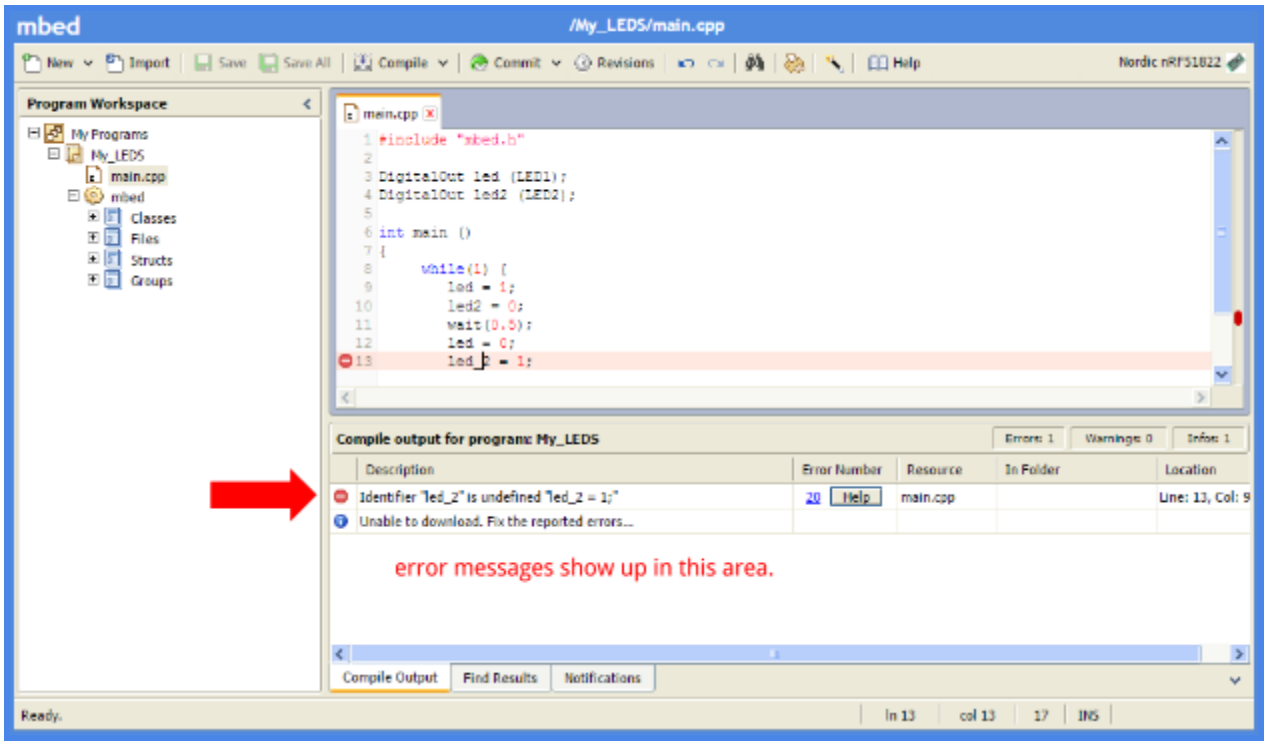
Success:

If you project does not have any issues it will compile successfully. This will be noted by a **green** arrow and the word “Success!” in the Console Output area at the bottom of the compiler screen. On a successful compile the browser will automatically try to download the program to your Downloads folder.



Error:

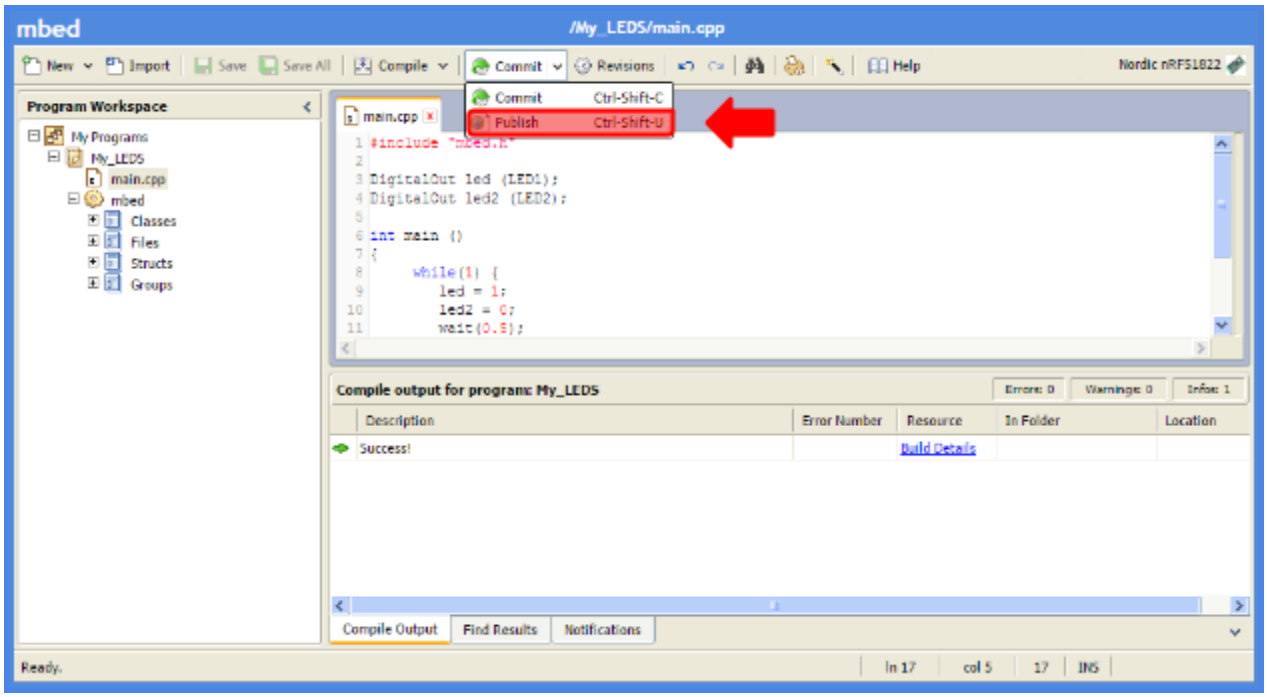
If there is a problem with the code then the issue will be described in the program output. To view where the problem is double click on the error description and the code editing tab will jump to and highlight the offending line of code.



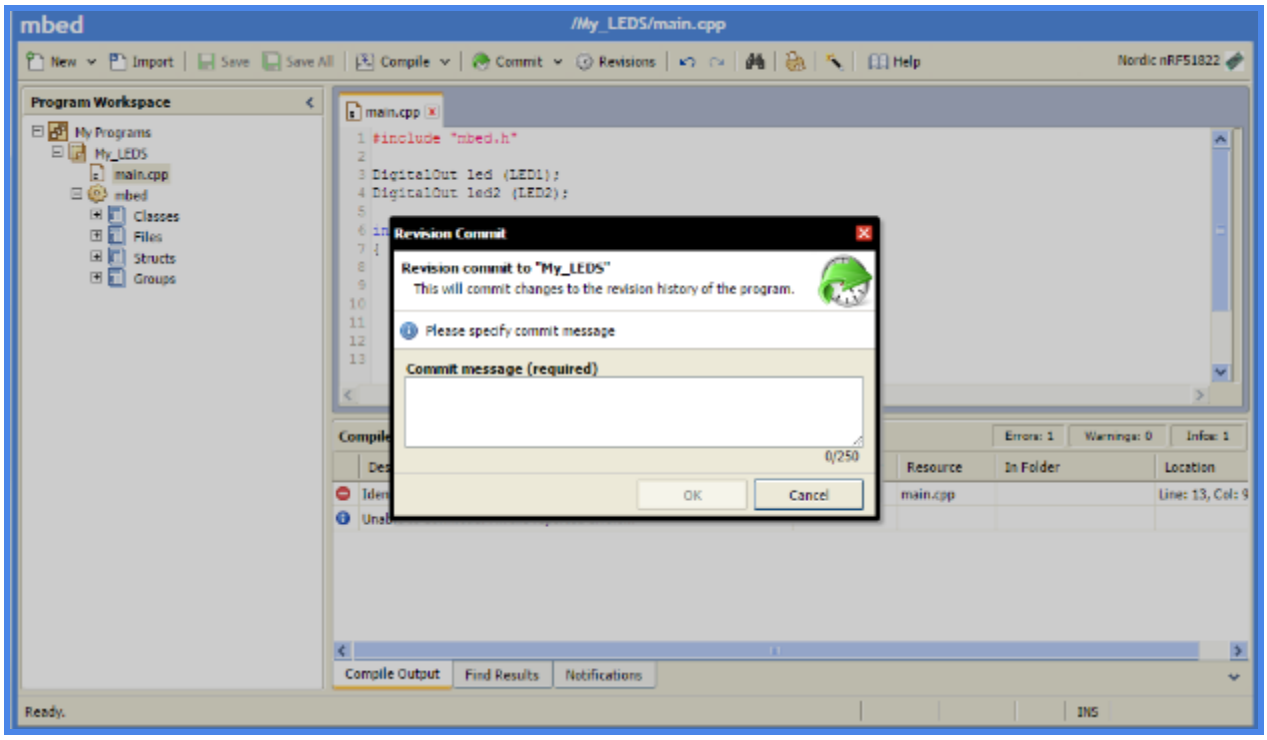
Publishing Code

When you are ready to share your code you can publish it to the mbed repository for other users to utilize.

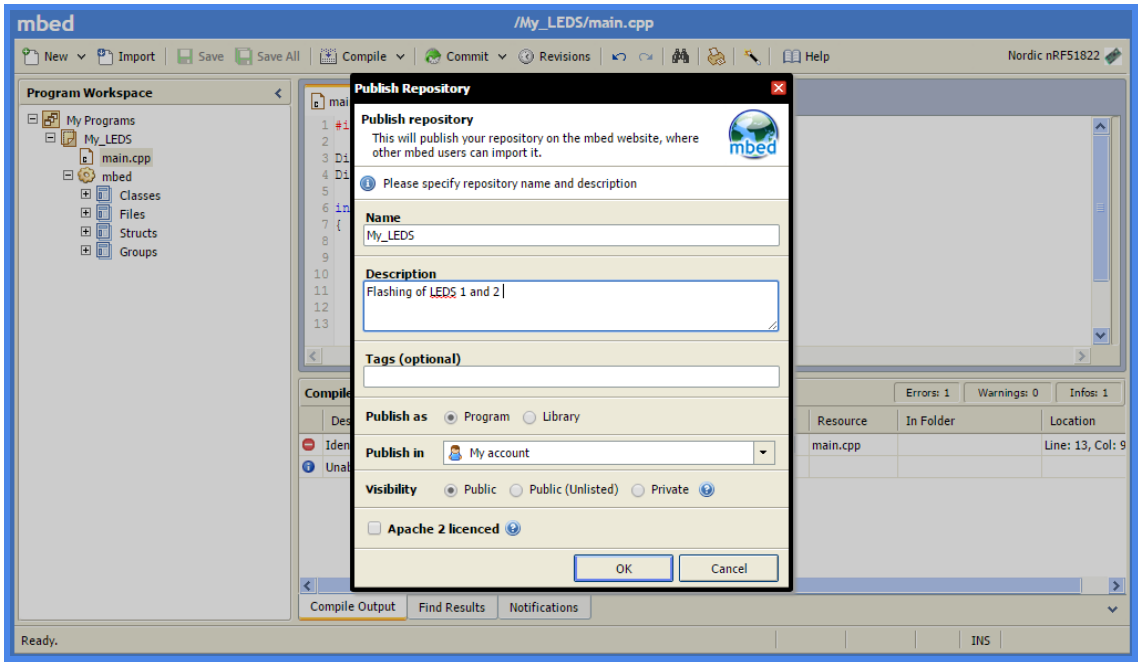
1. From the **Commit** drop-down list, select **Publish**.



2. The compiler will ask you to add a description of the revision history. You have up to 250 characters to describe the characteristics of the code, and you cannot commit the code without a description.

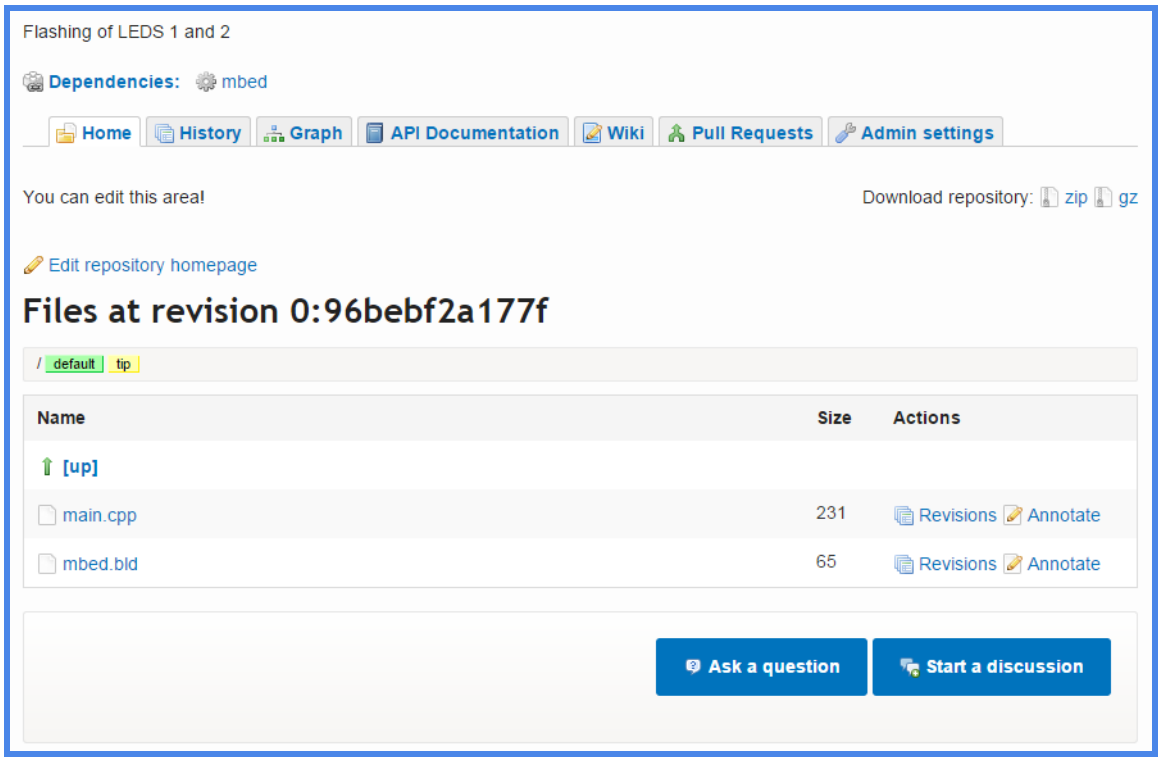


3. You can edit the program's name, description, tags and visibility.



4. Every published repository has a main page. You can edit the page, and are highly encouraged to add as much documentation here as you can, to make it easy for others to use and learn from your code.

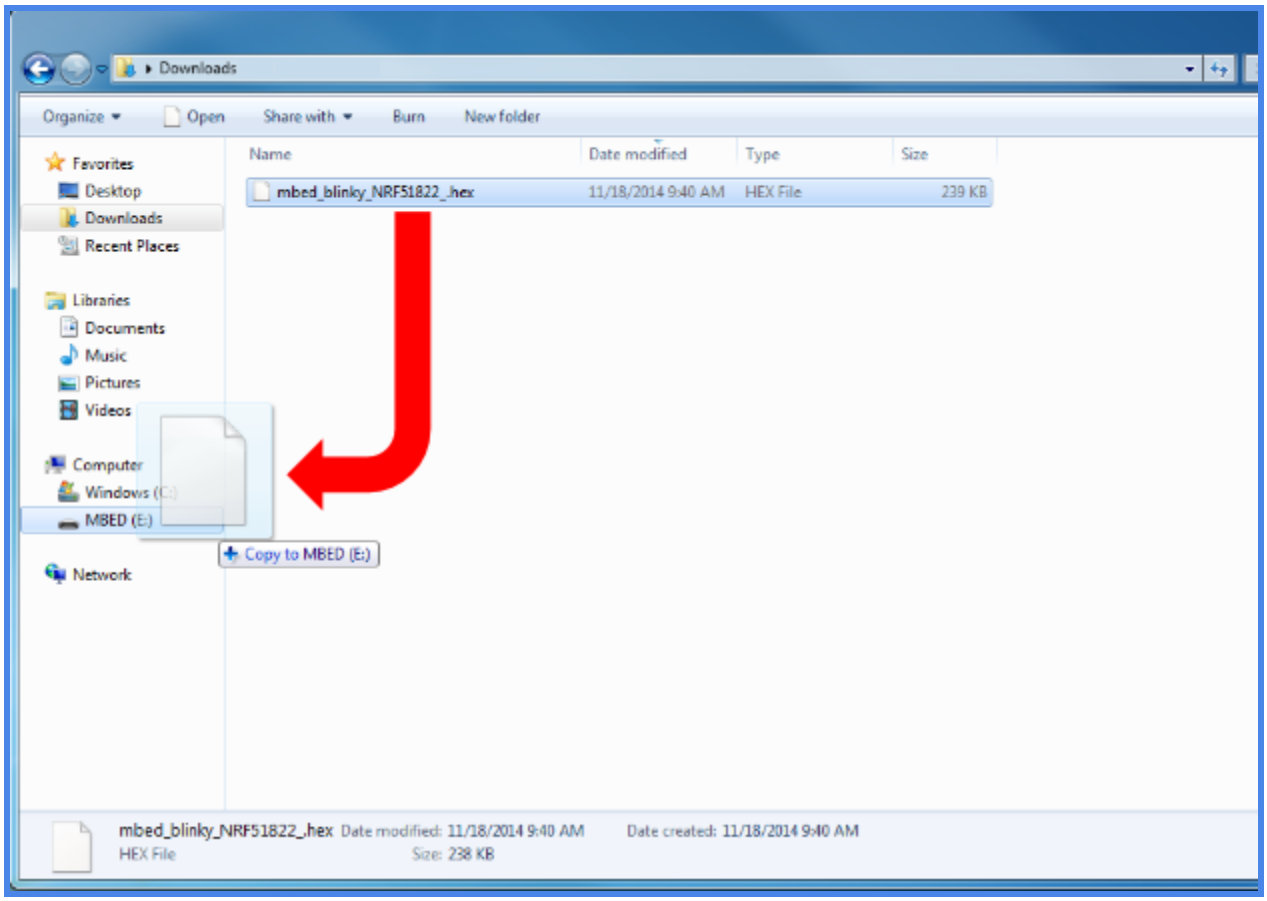
   Tip: See http://developer.mbed.org/cookbook/Wiki-Syntax for the markup syntax available.



Drag and Drop to the mbed Board to Program It

Once you have compiled your code and downloaded it to your computer programming the mbed board is a simple drag and drop.

1. Find the file generated by the compiler. By default, it should be in your Downloaded folder.
2. Drag the file to the mbed board.



3. Allow a couple of seconds for the mbed board to transfer the file. You should see a copy dialog and some lights flashing on the board. (On *nix systems you can also use the ‘cp’ command for the same effect).
4. Restart the board to initiate the code.

Success vs Failure : Ambulance Lights

Sometimes code will fail to run. This can either be an issue with the code being written incorrectly, or being compiled for the wrong board.

If the code fails in a spectacular way at the hardware level “ambulance lights” will flash on the board. This is to notify the user that something has spontaneously combusted. The most common reason is uploading code meant for one board onto another. This can be remedied by recompiling the code for the correct platform and reloading it onto the board.

Board Firmware Updates

If you have a problem with an out-of-the-box board, old firmware is a common culprit. To update the firmware for your board go to the Platform page, select your board and follow the instructions there.

Tip: If you are experiencing trouble connecting to your mbed board make sure to update both the target firmware and the interface chip firmware.

Collaboration

There are several things on the website that make collaborating on projects easy. The basic choice of publishing code as *private*, *public or public (unlisted)* allows for easy sharing. Posting notifications of cool projects to the projects page and on the forums is another great way of sharing projects.

Sometimes, though, there is a need for more advanced tools, like branching of code bases, sending pull requests and merging in changes. These advanced tools also exist on the developer.mbed.org website, but in a much more friendly way than on the terminal.
Basic Collaboration: http://youtu.be/BWM21JzSDSs
Advanced Collaboration: http://youtu.be/v0cgrNKhimY

Forking and Pull Requests

To make changes to existing code that someone else has published you need to import the code into your compiler, edit it and compile. When you publish your changes, rather than publishing normally you should select the *fork* option and publish the code to your own space. This will give the original code maintainer the opportunity to review the code.

Fork

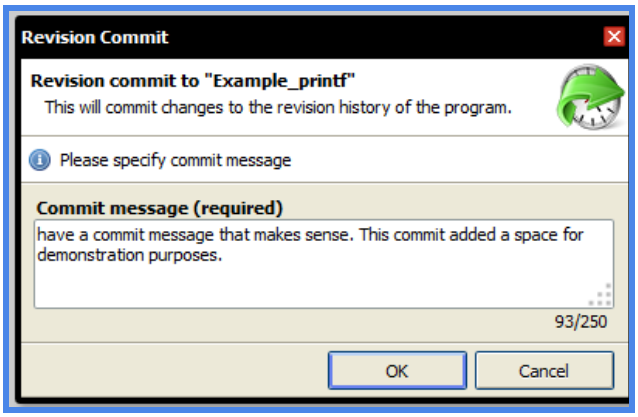
To fork the code on commit:

1. Make your changes and save your code.

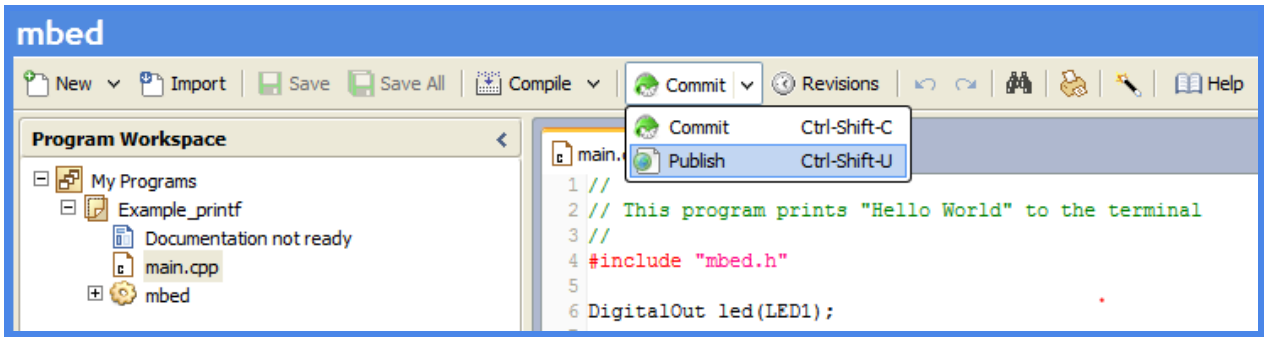
Tip: It is good practice to only change one very specific feature at a time. So if there are two separate issues to resolve, you should follow the procedure twice – once for each issue.

2. Commit your changes. From the **Commit** drop-down list, select **Commit.**

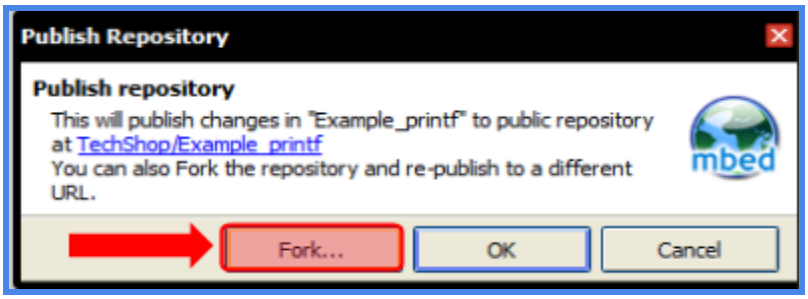
Tip: If you skip this step now, you'll have another chance to commit later in the publishing process.



3. Publish your changes (if you haven't committed them, you'll be given a chance to do it now). From the **Commit** drop-down list, select **Publish**.



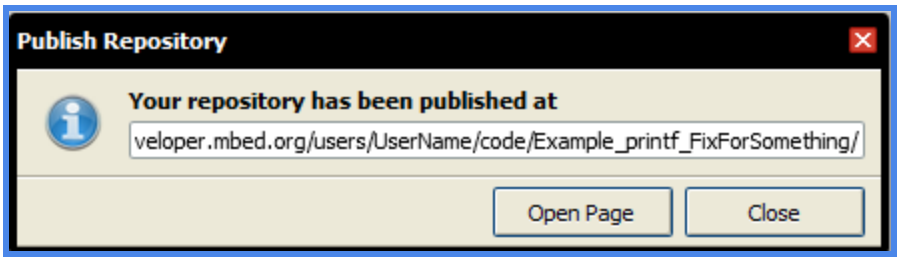
4. The compiler will ask you to confirm that you want to publish your code; click **Fork**.



5. Make sure you give the fork a meaningful name and description.

6. Publish the code as *public (unlisted)*. This will allow the original code maintainer to access the code and merge the changes while keeping search results clutter-free.

7. Click OK.
8. You will receive a publish link. This leads you to the repository's main page, where you can send a pull request (as explained below).



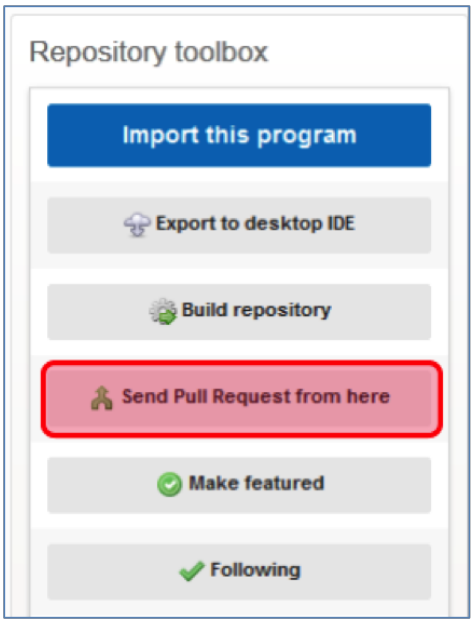
Pull Requests

Now that your changes have been published you should ask the original maintainer to add the changes to their code base. This is done through a method called a *pull request*.

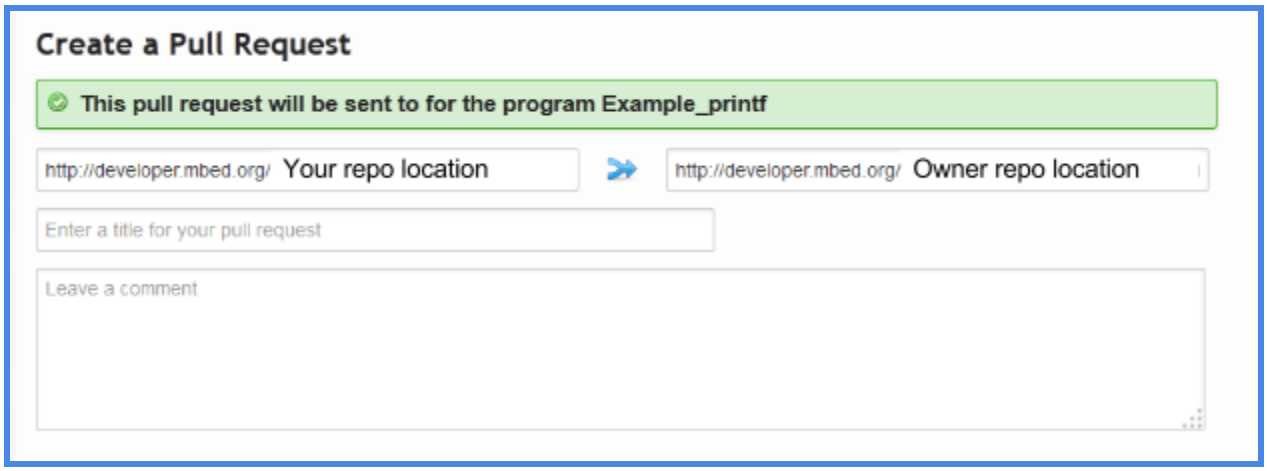
The pull request gives the maintainer the opportunity to accept or reject the changes you've made.

To make a pull request:

1. Go to the repository page (you got a link to it when you published the repository).
2. Click the **Send Pull Request From Here** button on the right-hand side of the page.
3. The request is sent to the code's maintainer.



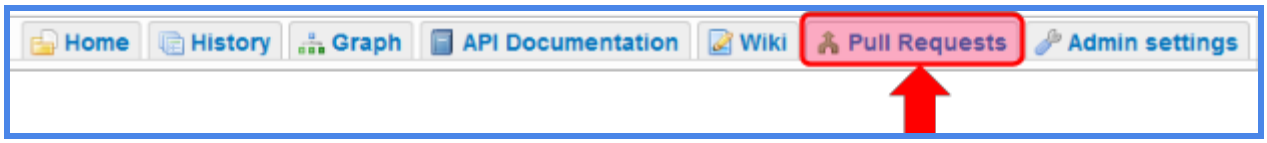
4. Fill out a description of the pull request. The more information you provide here, the better.



Managing and Merging Pull Requests

If you yourself maintain code, you may be sent pull requests. Pull request notifications will show up on your dashboard, as well as on the repository page for that code.

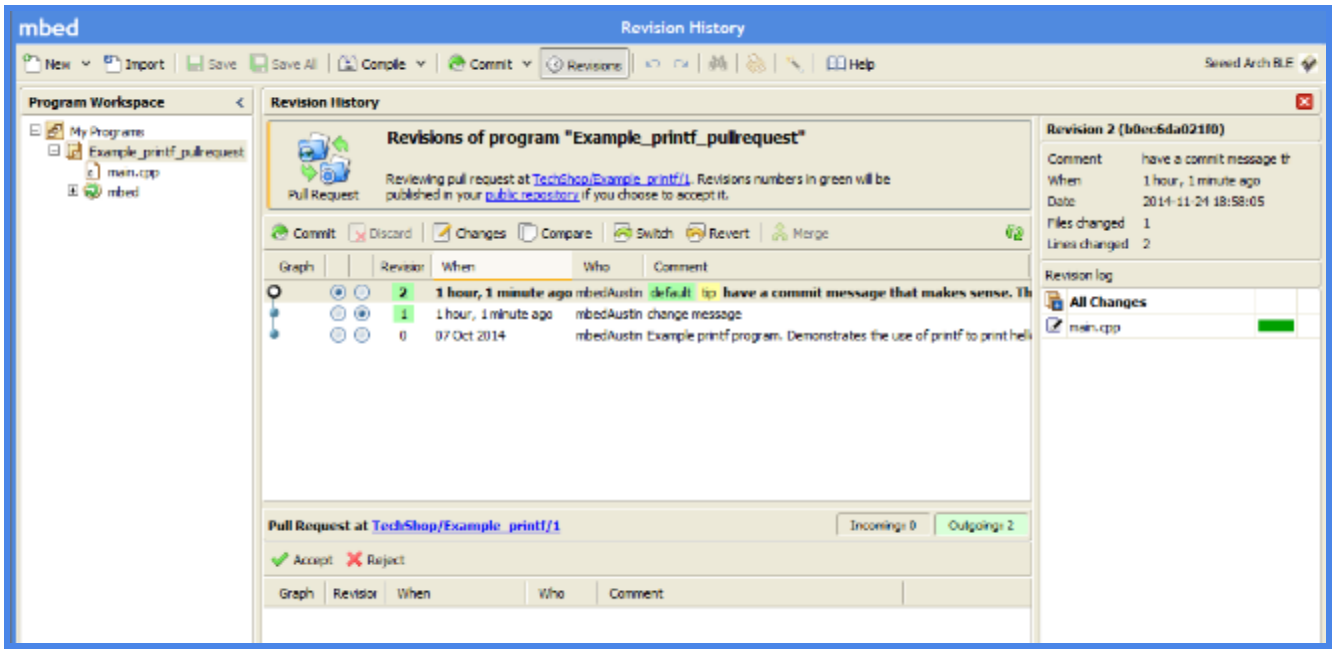
Tip: You can see all pull requests open against a code repository by looking at the **Pull Requests** tab.



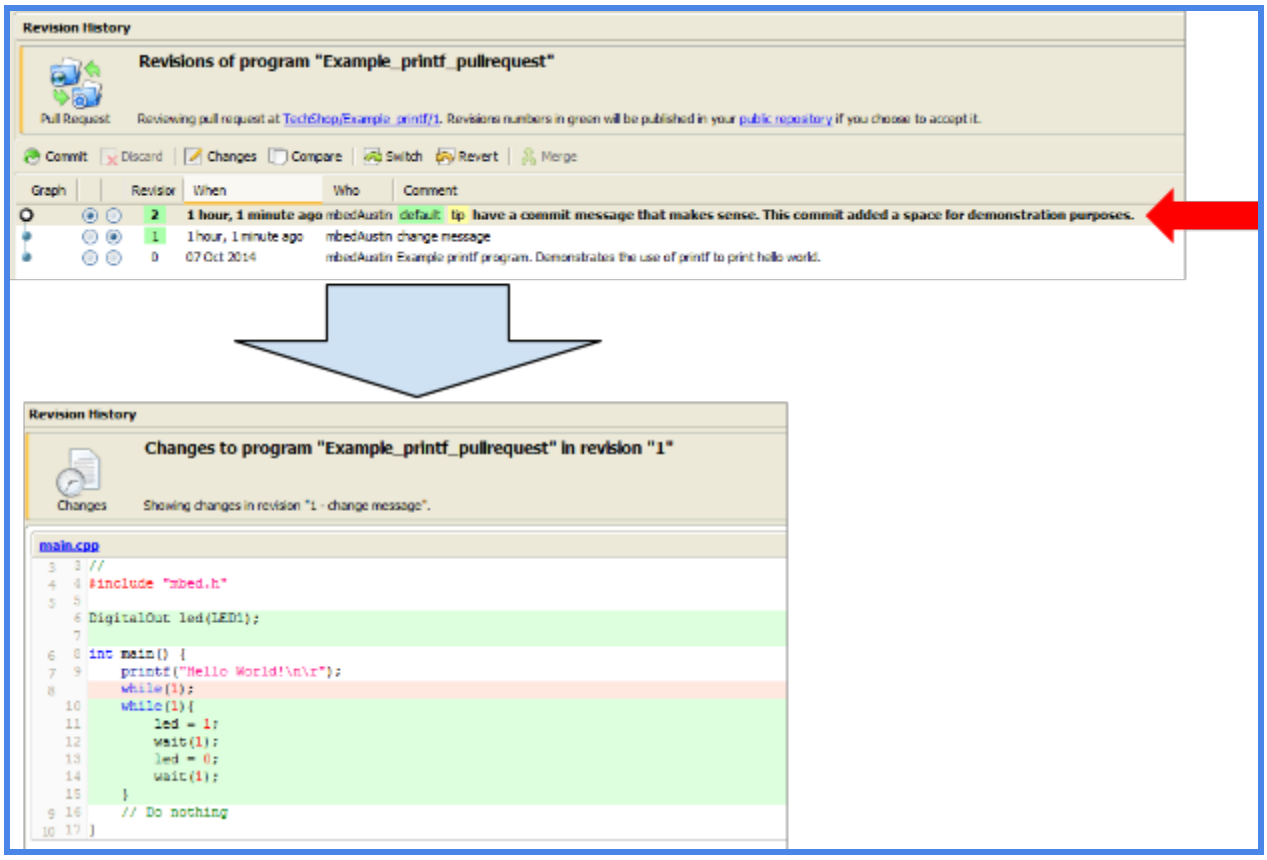
To review the changes:

1. Click the **Review** button. The code will be imported into your compiler.

2. The compiler's default view will be of the revisions made to the code.

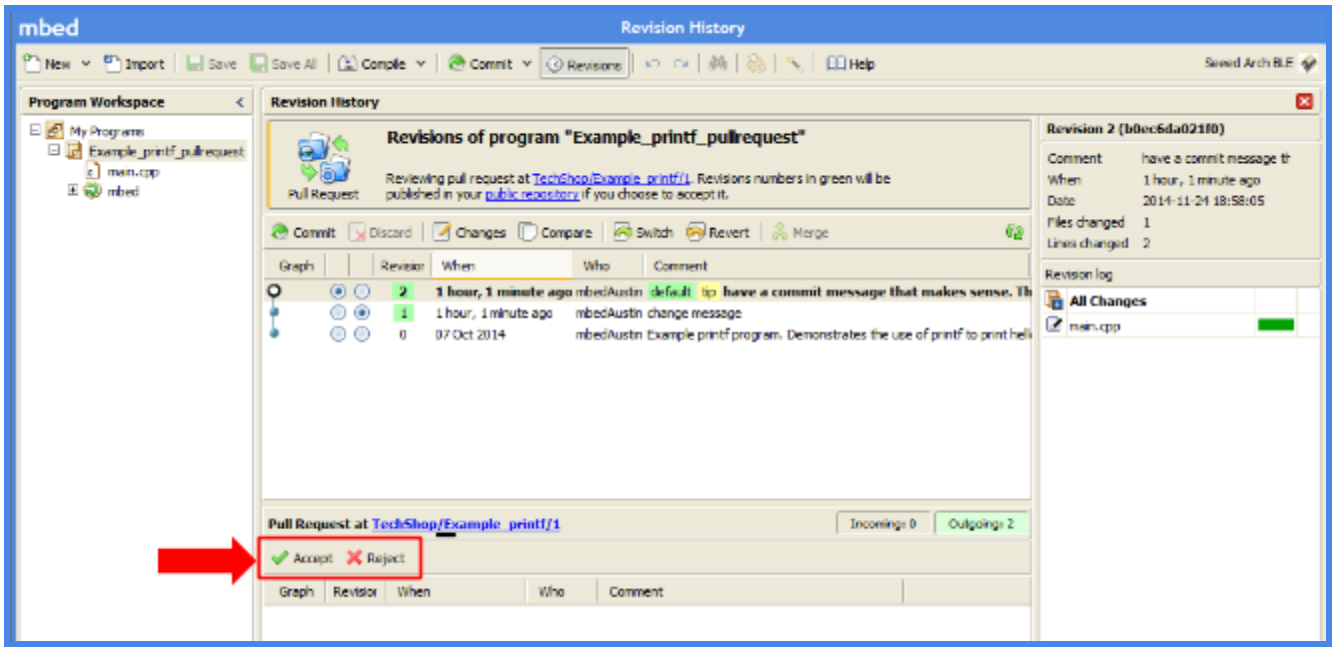


3. To view the differences between the code commits double click on the comment.
4. A diff view opens, highlighting the differences in the code.



5. To accept or reject the changes click **Accept** or **Reject** on the main Revisions page:
    a. If you accept the pull request, the changes are merged into the main code branch.
    b. If you reject the pull request, the changes are kept out of the main.

Tip: It is highly recommended you provide feedback on the pull request, especially if you reject it, so the person on the other side can make changes and resubmit the pull request.


mbed
Revision History
New
Import
Save
Save All
Compile
Commit
Revisions
Help
Program Workspace
My Programs
Example_printf_pullrequest
main.cpp
mbed
Revision History
Revisions of program "Example_printf_pullrequest"
Pull Request
Reviewing pull request at TechShop/Example_printf/1. Revisions numbers in green will be published in your public repository if you choose to accept it.
Commit
Discard
Changes
Compare
Switch
Revert
Merge
Graph
Revision
When
Who
Comment
2
1 hour, 1 minute ago
mbedAustin
default
tip
have a commit message that makes sense. Th
1
1 hour, 1 minute ago
mbedAustin
change message
0
07 Oct 2014
mbedAustin
Example printf program. Demonstrates the use of printf to print hell
Pull Request at TechShop/Example_printf/1
Incoming: 0
Outgoing: 2
Accept
Reject
Graph
Revision
When
Who
Comment
Revision 2 (b0ec6da02180)
Comment
have a commit message th
When
1 hour, 1 minute ago
Date
2014-11-24 18:58:05
Files changed
1
Lines changed
2
Revision log
All Changes
main.cpp