# Creating a Makefile project

This tutorial describes the process of creating a new C++ project that includes a makefile, and debugging the program.

You need to create a project to contain your source code and related files. A project has an associated builder that can incrementally compile source files as they are changed.

To create a project:

1. Select **File > New > Project**.  
     
   When you create a new project, you are required to specify the project type. This project type will determine the toolchain, data, and tabs that the CDT uses/displays.
2. Select the type of project to create. For this tutorial, expand the **C/C++** folder and select **C++ Project**.  
     
   The **C++ Project** wizard opens. Click here to see an illustration.  
     
   By default, the CDT filters the **Toolchain** and **Project types** that currently display in those lists are based on the language support for the C++ Project wizard you selected for this tutorial.
3. In the **Project name** field, type HelloWorld.
4. Leave the **Use Default Location** option selected.  
     
   Next, you want to select the type of project to create. In the New CDT Project Wizard, you can choose from the following project types:
   * **Executable** - Provides an executable application. This project type folder contains three templates.  
     + **Hello World C++ Example** provides a simple C++ Hello World application with main().
     + **Hello World ANSI C Example** provides a simple C Hello World application with main().
     + **Empty Project** provides a single source project folder that contains no files.  
       After you select this template, the result is a project with only the meta-data files required for the project type. You are expected to provide source files for the project's target.

The makefile for the **Executable** project type is automatically created by the CDT.

* + **Shared Library** - An executable module that is compiled and linked separately. When you create a project that uses a shared library (libxx.so), you define your shared library's project as a Project Reference for your application. For this project type, the CDT combines object files together and joins them so they're relocatable and can be shared by many processes. Shared libraries are named using the format libxx.so.version, where version is a number with a default of 1. The libxx.so file usually is a symbolic link to the latest version. The makefile for this project type is automatically created by the CDT.
  + **Static Library** - A collection of object files that you can link into another application (libxx.a). The CDT combines object files (i.e. \*.o) into an archive (\*.a) that is directly linked into an executable. The makefile for this project type is automatically created by the CDT.
  + **Makefile Project** - Creates an empty project without the meta-data files. This selection is useful for importing and modifying existing makefile-based projects; a new makefile is not created for this project type.

By default, the CDT filters the **Toolchain** and **Project types** that currently display in those lists based on the language support for the C++ Project wizard you selected for this tutorial.

1. From the **Project types** list, expand **Makefile project** and select **Empty Project**. This project lets you enter the source file and the makefile yourself.
2. Select a required toolchain from the **Toolchain** list.  
   A toolchain is a set of tools (such as a compiler, linker, and assembler) intended to build your project. Additional tools, such as a debugger, can be associated with a toolchain.  
   There can be several toolchains available, depending on the compilers installed on your system.
3. Click **Next**.
4. (Optional) On the **Select Configurations** page, Click **Advanced Settings**.
   * Expand **C/C++Build > Settings**. Click the **Binary Parsers** tab.  
       
     Click here to see an illustration.
   * In the **Binary Parser** list, select the error parser that is appropriate for your platform. Other project settings can be modified here as well. The default selection is probably OK.

To ensure the accuracy of the C/C++ Projects view and the ability to successfully run and debug your programs, selecting the correct parser is important. After you select the correct parser for your development environment and build your project, you can view the components of the .o file in the C/C++ Projects view. You can also view the contents of the .o file in the C/C++ editor.Click **OK** to close the Project Properties window.

1. Click **Finish**.
2. If a message box prompts you to change perspectives, click **Yes**.

Your new project displays in the Project Explorer view. Your project is empty because you have not yet created files for your project. You may see an error since there is nothing to build yet for your project. You can now start writing the code for your HelloWorld program.

**Tip:** You can view and modify the properties of your HelloWorld project at any time by right-clicking on the project in the **Project Explorer** view and clicking Properties.

  [**Next: Creating your C++ file**](http://docs.google.com/cdt_w_newcpp.htm)

For more information about:

* Projects, see **Workbench User Guide > Concepts > Workbench > Resources**
* The workspace, see **Workbench User Guide > Tasks > Upgrading Eclipse**

****

[CDT overview](http://docs.google.com/concepts/cdt_c_over_cdt.htm)

[CDT projects](http://docs.google.com/concepts/cdt_c_projects.htm)

[Project file views](http://docs.google.com/concepts/cdt_c_proj_file_views.htm)



[Working with C/C++ project files](http://docs.google.com/tasks/cdt_o_proj_files.htm)



[C/C++ Projects view](http://docs.google.com/reference/cdt_o_proj_prop_pages.htm)

