# CDT projects

Before you can work in the CDT, you must create a project to store your source code, makefiles, binaries, and related files. C/C++ projects are displayed in the C/C++ Projects view.

**Tip:** Nested projects are not supported. Each project must be organized as a discrete entity. Project dependencies are supported by allowing a project to reference other projects that reside in your workspace. For more information, see [Selecting referenced projects](http://docs.google.com/reference/cdt_u_prop_general_pns_ref.htm).

For more information about projects and where they are stored, see:

* **Workbench User Guide > Tasks > Resources**
* **Workbench User Guide > Tasks > Running Eclipse**

## Project types

You can create a C or C++ project.

To build your project, you can either create your own makefile, or let CDT generate the makefiles for you automatically.

You can toggle these modes (generate makefiles or not) at any time for existing project. Use [properties.](http://docs.google.com/reference/cdt_o_proj_prop_pages.htm)

When you create a new project, you are required to specify the project type. This project type will determine the toolchain and data, and tabs that the CDT uses. 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 a template, the result is a project with only the meta-data files required for the project type. You are expected to modify these source files, as required, and 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.

## Project conversion

You can convert projects from C to C++ (or from C++ to C). If, for example, your requirements change and you must convert an existing C project to C++, you can do this without recreating the project. The CDT converts your project files and resolves any source control issues.

## A few notes about projects

* When you create a file within a project, a record (local history) of every change is created. For more information about local history, see **Workbench User Guide > Reference > User interface information > Development environment > Local history**.
* Spaces in projects and filenames can cause problems with some tools, such as the make utility or the compiler.
* Be careful when you use only case to distinguish files and projects. UNIX-based operating system file names are case sensitive, but Windows filenames are not. Therefore, Hello.c and hello.c are separate files in UNIX but overwrite each other in Windows.

For more information about projects, see **Workbench User Guide > Concepts > Workbench > Resources**.



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

[How to bring C/C++ source into Eclipse](http://docs.google.com/getting_started/cdt_w_existing_code.htm)



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

[Converting a C or C++ nature for a project](http://docs.google.com/tasks/cdt_t_conv_proj.htm)



[Project properties](http://docs.google.com/reference/cdt_o_proj_prop_pages.htm)

[Views](http://docs.google.com/reference/cdt_o_views.htm)

