## Tutorial: Importing an existing project

The following tutorial takes you though the process of importing a simple application using the CDT.

Step 1: You want to create a single project that will reflect all of the components for the existing source tree.

1. Select **File > New > Project**.
2. For this tutorial, expand **C++** and select **C++ project**.  
     
   The **New Project** wizard displays. Click here to see an illustration (displayed in a separate window).
3. In the **Project name** field, type a name for the project.
4. Ensure that the **Use default location** option is not selected because you will need to specify where the resources reside in the file system since they do not reside in your workspace.
5. In the **Location** field, specify a path for your project.
6. From the **Project types** list, expand **Makefile Project** and select **Hello World C++ Project**.
7. Click **Next**.
8. Select only the configurations you want created. Configurations display for each toolchain selected on the previous page of the wizard.
9. Click **Finish** to close the dialog box.  
   You will see the new project in the C/C++ Projects view. In addition, new 'dot' files in your legacy project's root directory, these are CDT project files.

Step 2: You are now ready to build your project.

1. To build your project, select **File** > **Build Project**.  
   After the project build completes, the results display in the **Console** view and new objects, such as binaries and includes, show in the **Projects** view.
2. Review the results to ensure that there are no errors.  
   **Note:** By default, the indexer is set to **Fast indexer** for the project to parse your project in the same way that a compiler does; beginning with each compilation unit and parsing that file and all files that it includes, except that it parses each header file included in a project only once. This method provides the most accurate index information. For large projects using complex C++ code, this indexer can be slow. For example, if a header file is included and takes two compilation units, the parsing of the second unit reuses the results of parsing the first unit. This is similar to how precompiled headers work. The indexing of large projects using the Fast indexer uses fewer resources than the Full indexer, but the resulting index is not quite as accurate.

Step 3: You are now ready to run your application.

1. To run your application within the C/C++ Perspective, click **Run > Run**  
     
   The **Run Configurations** dialog displays.
2. To create a run configuration, select **Local C/C++ Application** and click **OK**.
3. Select the new run configuration in the **Configurations** view and click the **Main** tab to edit your run configuration.
4. Optional: You can change the name of your run configuration by entering a new name in the **Name** field.
5. To select the project, click **Browse** opposite the **Project** field.
6. From the Project Selection window, select the project and click **OK**.
7. Select the new **C/C++ Application** you want to execute by clicking **Search**.
8. From the Program Selection window, select an appropriate binary from the list and click **OK**.
9. Click **Apply** to save the run configuration and then click **Run**.  
   The application runs in the **Console** view. The **Console** also shows which application it is currently running in the title bar. You can configure the view to display different elements, such as user input elements using different colors.

