

Creating and using a static library in eclipse

This document the details of setting up a static library in eclipse and using it in a test project.

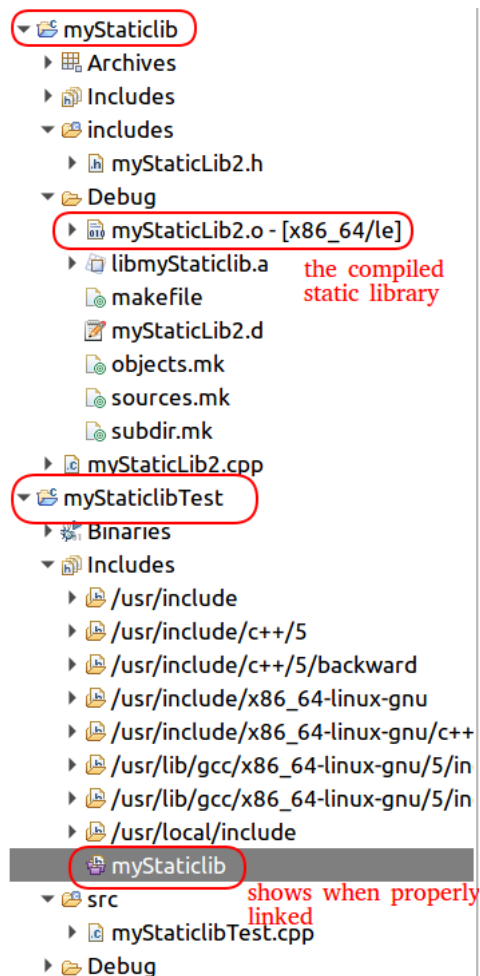
Projects Created:

myStaticlib

myStaticLibTest

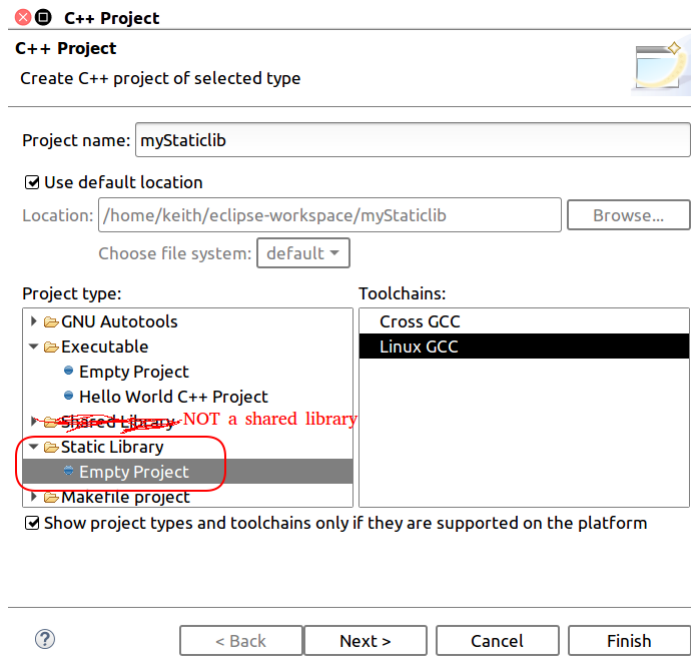
Completed Eclipse Directory Structure

Here is how eclipse's directory structure will look when done



Create The **Static library** project (**myStaticlib**)

1. Create static library project.

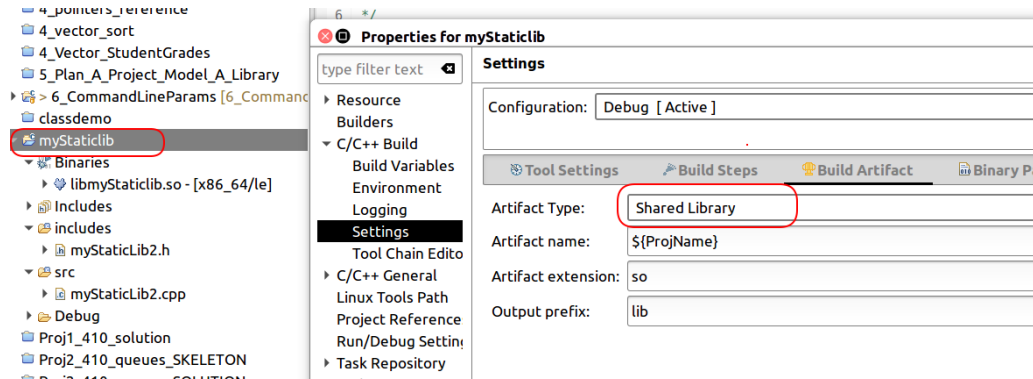


2. Add a src folder to the project (not a plain folder)
3. Add a cpp and corresponding .h file. Add a function (**no main()**) to this library in both the cpp and header. For instance;

```
Project2.cpp  myStaticlibTest.cpp  myStaticLib2.cpp
1  /*
2   * myStaticLib2.cpp
3   *
4   * Created on: Oct 6, 2017
5   * Author: keith
6   */
7  #include "./includes/myStaticLib2.h"
8
9  #include <iostream>
10
11 void show(){
12
13     std::cout<<"In library"<<std::endl;
14 }
15
```

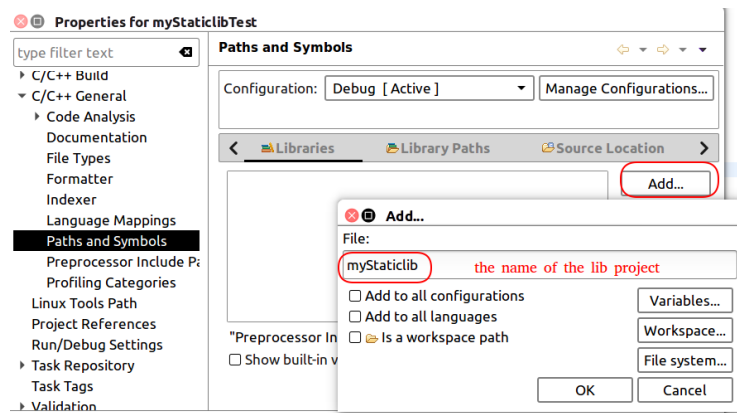
```
myStaticLib2.h
1  /*
2   * myStaticLib2.h
3   *
4   * Created on: Oct 6, 2017
5   * Author: keith
6   */
7
8  #ifndef MYSTATICLIB2_H_
9  #define MYSTATICLIB2_H_
10
11 void show();
12
13 #endif /* MYSTATICLIB2_H_ */
```

4. Compile it. The library will have the prefix 'lib' and suffix '.so'. You can look at its properties to verify that it's a static library (select project, Properties, then in the following UI)

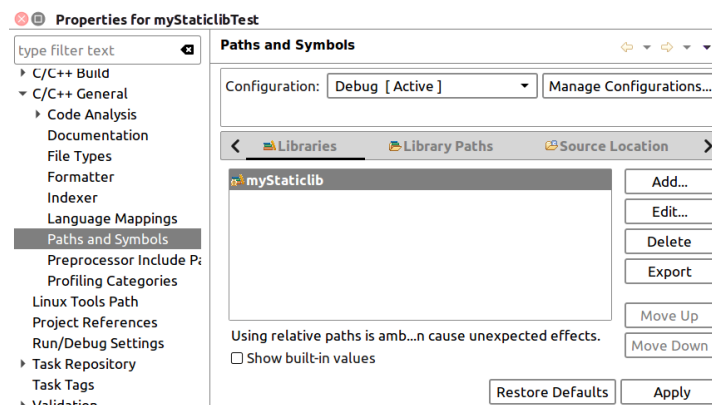


Create a **test project** that uses the static library (**myStaticLibTest**)

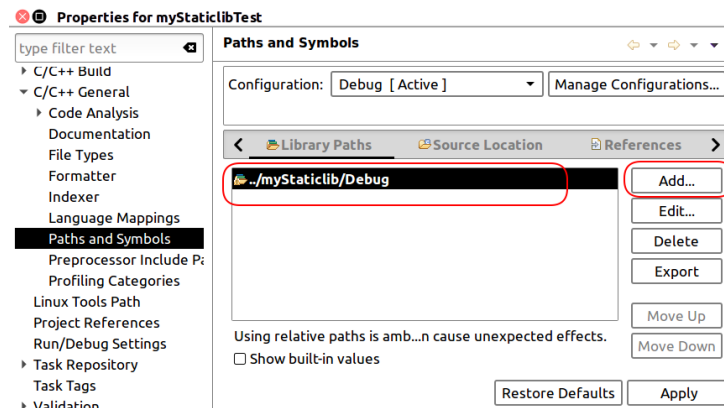
5. Create a regular project.
6. Add your library to this project. Click on your regular project, then properties, then C/C++ General, Paths and Symbols, then the Libraries tab(see the illustration below). Click the 'Add' button and follow the image below .



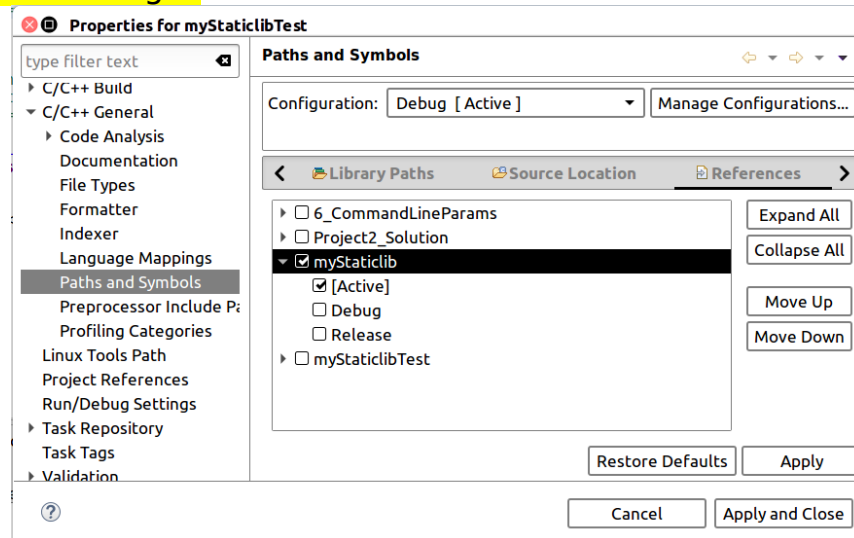
When done your library path should appear in the text window like so;



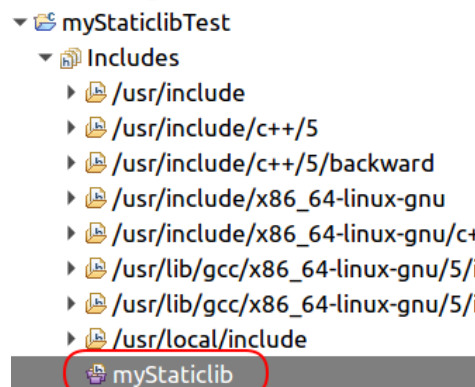
7. Add a path to look for this library



8. Finally select the reference tab and add a reference to the library you want to link to this application. **Be sure to click the Apply button to save your changes**



9. At this point your lib should appear in the includes directory of your project.



10. Include the libraries header in the test file. If you did not do step 7 correctly, this will not work. And add a call to the library function.



The image shows a code editor window with a file named `myStaticLibTest.cpp`. The code is as follows:

```
1 // =====
2 // Name      : myStaticLibTest.cpp
3 // Author    :
4 // Version   :
5 // Copyright : Your copyright notice
6 // Description : Hello World in C++, Ansi-style
7 // =====
8
9 #include <iostream>
10 using namespace std;
11 #include "../myStaticLib/includes/myStaticLib2.h"
12
13 int main() {
14     show();
15 }
```

Annotations in the image include:

- A red circle around the filename `myStaticLibTest.cpp` in the tab.
- A red circle around the `#include` statement on line 11.
- A red circle around the `show();` function call on line 14.
- The `using namespace std;` line on line 10 is crossed out with a red line.