Simple C Tutorial

Creating a Project

In order to help you become acquainted with C-Forge we will lead you through a simple project; let's call it "First". When "First" is completed, it will print the line "Hello, World!".

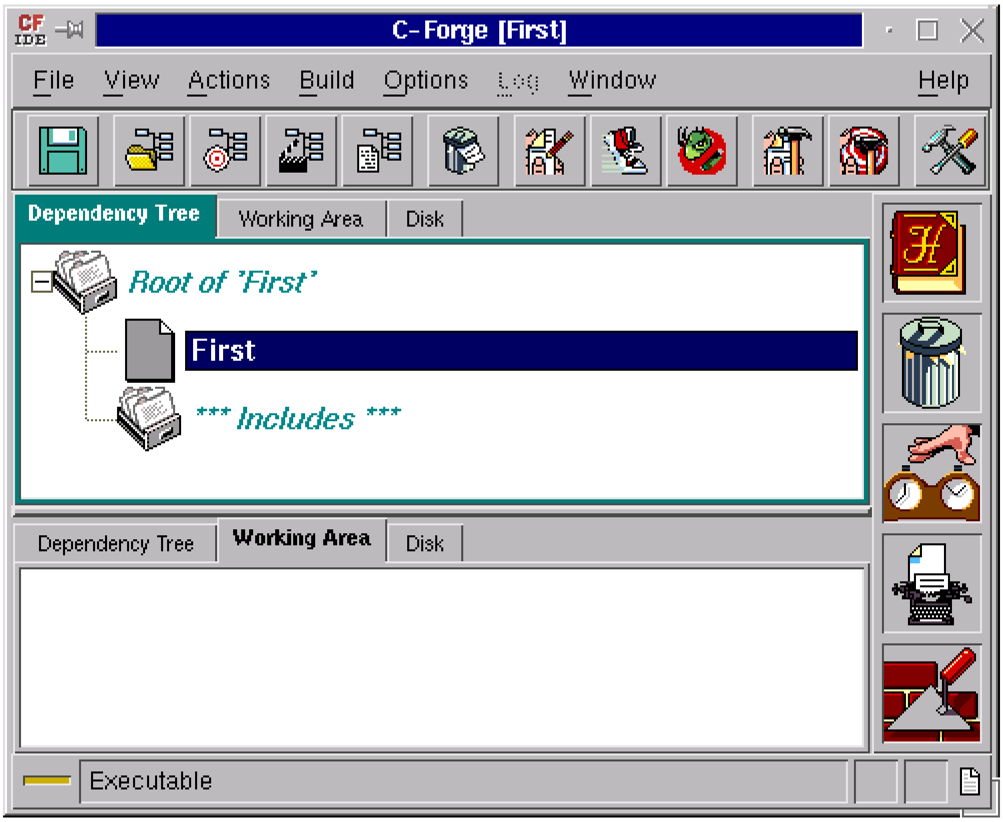
Click the New Project toolbar button found in the C-Forge main window.

This opens the New Project dialog box. This dialog already contains the Project Root /home/user.

Type in the name "First" and click the OK button. A new directory called /home/user/First-prj has been created containing the the project file First.mak. In addition the Create new target(s) dialog is opened.

To create a "C" program, choose the "C Executable" target type from the list (C-Forge also supports a large variety of other languages). Enter the name for the first target of the project. Click the OK button. Since we decided to keep the default directory structure C-Forge has created a directory called /home/user/First-prj/First-dir with subdirectories Sources, Objects and Includes.

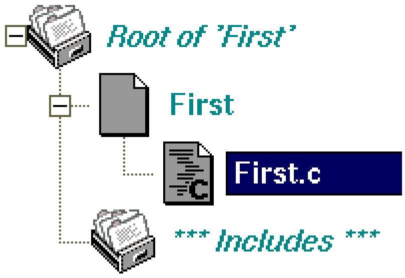
Your Project Desktop window looks like this:



Adding Source

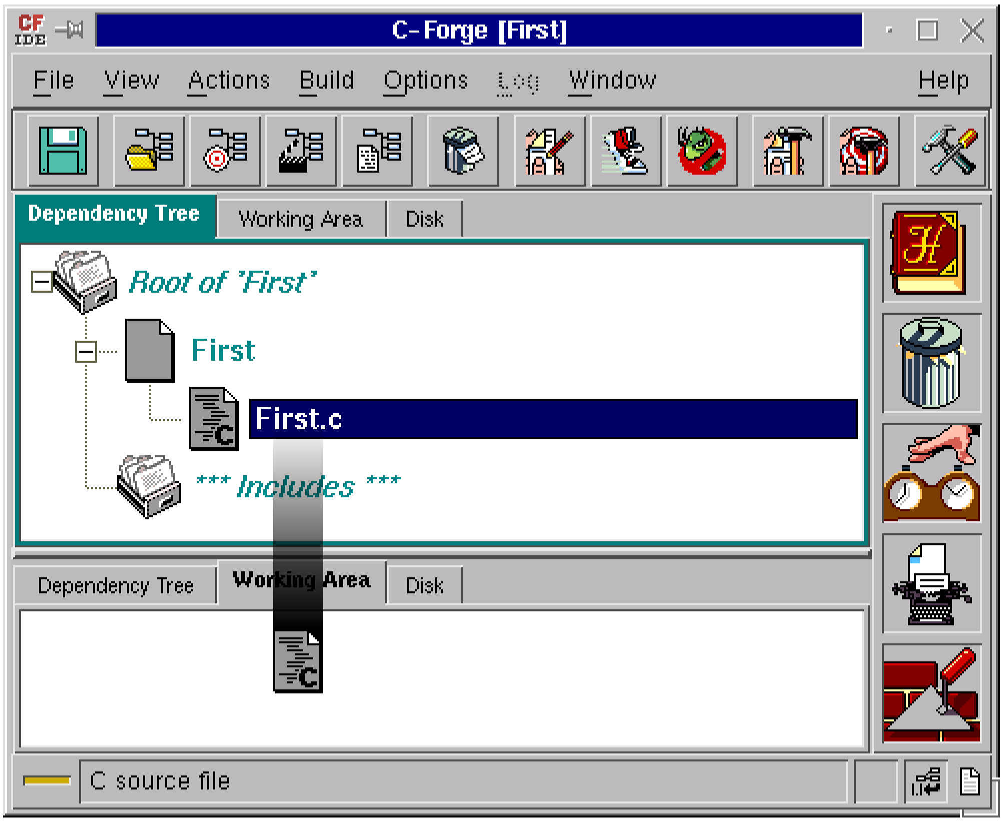
Choose the Add Source button, found in the Project Desktop toolbar to create a source file for the target. This will open the Create new node(s) Dialog.

Type the source file name - "First.c" and click the OK button (Make sure that type of this source is "C source file" and check-box "Add as include" is disabled). We have previously selected the default directory layout so the new file is created in the /home/user/First-prj/First-dir/Sources directory. If we had checked the Revision Control check-box when we created this file C-Forge would have created an RCS directory with an initial depository of 'First.c,v'. The dependency tree now looks like this:



Editing Source

We are now ready to modify "First.c". Using the middle mouse button drag "First.c" from the Dependency Tree to the Working Area or Edit drop site.



If the Revision Control option had been selected the *Check-Out* operation is automatically performed.

- The default version is extracted from the RCS depository and locked.
- That version is located and the file is copied to the users working

  diretory.
- The file is placed in the editor window.
- If the Insert RCS header option was checked, in the Create new node(s) Dialog during the file creation step, RCS information is added to the top of the file.

If the Revision Control option was not selected the file is copied to the the user working directory and placed in the editor.

Type the following "C" code into the opened Editor Window:

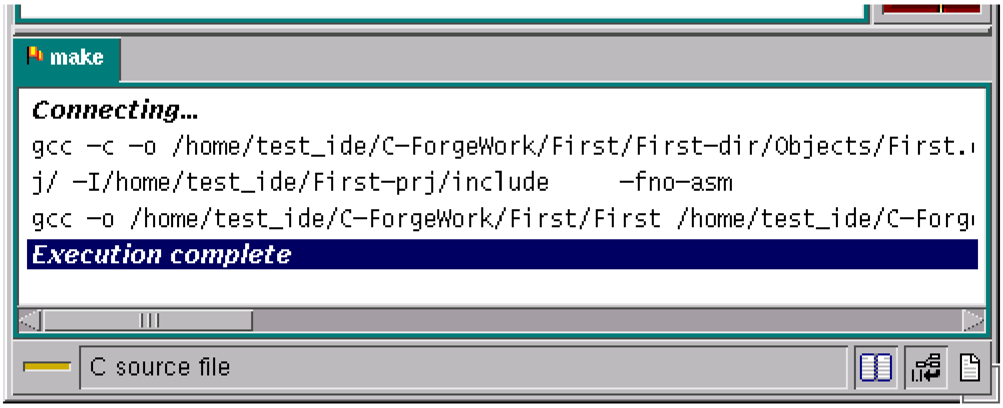
```
int main (void)
{
????? printf ("Hello, World!\n");
????? return 1;
}
```

Save changes in the editor.

Making Target

Select the Actions -> Make Target option found in the Editor Window menu. This will have the following affect:

- The project log window is opened,
- "First.c" file is compiled to "First.o", and "First.o" file is linked to "First" executable file:



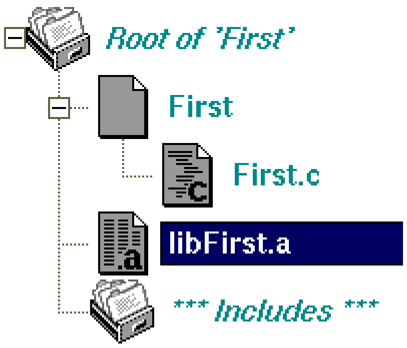
```
Connecting...
gcc -c -o /home/test_ide/C-ForgeWork/First/First-dir/Objects/First.
j/ -I/home/test_ide/First-prj/include     -fno-asm
gcc -o /home/test_ide/C-ForgeWork/First/First /home/test_ide/C-Forg
Execution complete
```

If no errors have been found, select the Actions -> Run Target from Editor menu to execute target "First".

Adding a Library to Project

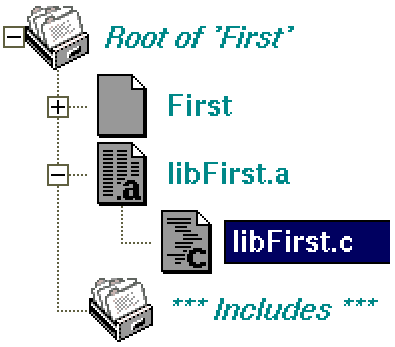
Now that we have a simple project ("First") up and running we are ready to add a library. For the purposes of this exercise we will name the library "libFirst".

1. From the Project Desktop press the Add Target button located on the toolbar. This will open the Create new target(s) dialog.
2. Choose "Library" item from the Type list and type in the library name "libFirst.a" in the target field. Once that is done click the OK button to continue. The Dependency Tree look like this:



?

3. It is now time to add a source module to our library. To do this click the Add Source button found in the toolbar. This will open the Create new node (s) dialog.
4. Type the file name "libFirst.c". The Dependency Tree now looks like this:



?

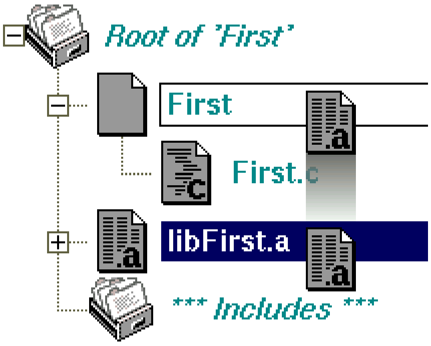
5. Using the middle mouse button drag "libFirst.c" from the Dependency Tree to the Working Area or Edit drop site.
6. Next create the text for "libFirst.c" as we did it above.

```
void libFirst (char *string)
{
?????? printf (string);
}
```

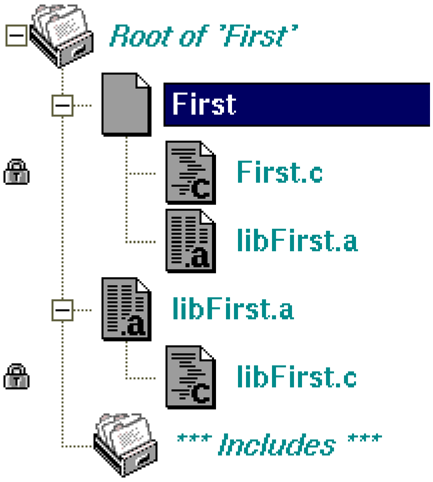
7. Select the *Actions -> Make Target* option found in the Editor Window. This will have the following affect:
   - "libFirst.c" file will be compiled to "libFirst.o" file,
   - "libFirst.a" is created containing "libFirst.o".
8. Now that we have built the library "First.c" needs to be modified to include a call to "libFirst". Drag "First.c" from the Dependency Tree to the Working Area or Edit drop site and make the following changes:

```
extern void libFirst (char *);
int main (void)
{
????? libFirst ("Hello, World!\n");
????? return 1;
}
```

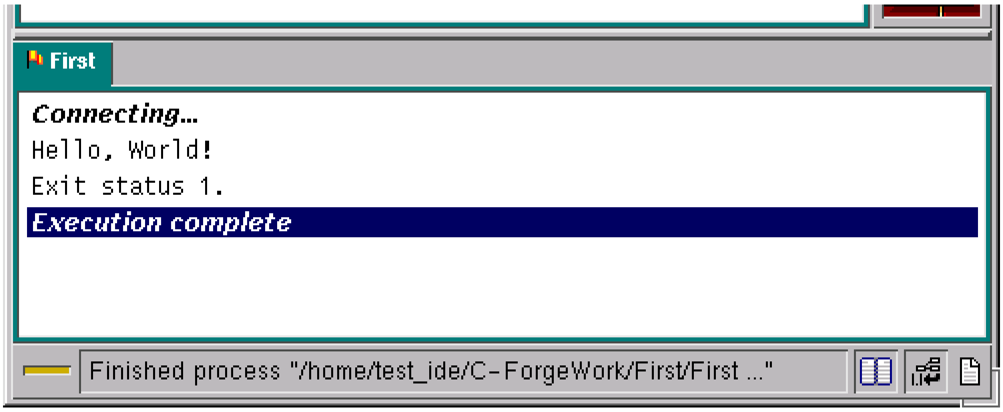
9. All that remains is to link the library to our target. To do this, Control-Shift-Drag'n'Drop (link) the target "libFirst.a" onto target "First".



And the full dependency tree looks like this:



10. Now, Drag'n'Drop target "First" to Make drop site to compile and link it.
11. Finaly double click on "First" to run it.



```
Connecting...
Hello, World!
Exit status 1.
Execution complete
```

Finished process "/home/test_ide/C-ForgeWork/First/First ..."