How to Build Your Own Library in NetBeans

Step 1: Start a new project for the library

Choose C/C++ Static Library, and an appropriate name for the project

Step 2: Add/create the code and header files (*.c and *.h), and then build the library

This will produce a static library called libXXXX.a (or libXXXX.dll on Windows), which will be located in the dist/Debug/<OS Architecture> folder under your library project folder.

Example:

Library project name = plot
OS Architecture = GNU-MacOSX

Output of build command:

```
mkdir -p build/Debug/GNU-MacOSX
.
.
.
rm -f "build/Debug/GNU-MacOSX/gplot_theory.o.d"
gcc    -c -g -MMD -MP -MF "build/Debug/GNU-MacOSX/
```

```
gplot_theory.o.d" -o build/Debug/GNU-MacOSX/
gplot_theory.o gplot_theory.c
.
.
.
mkdir -p dist/Debug/GNU-MacOSX
rm -f dist/Debug/GNU-MacOSX/libgplot.a
ar -rv dist/Debug/GNU-MacOSX/libgplot.a build/Debug/
GNU-MacOSX/gplot_basic.o build/Debug/GNU-MacOSX/
gplot_errors.o build/Debug/GNU-MacOSX/
gplot_errorstheory.o build/Debug/GNU-MacOSX/
gplot_theory.o
ar: creating archive dist/Debug/GNU-MacOSX/libgplot.a
a - build/Debug/GNU-MacOSX/gplot_basic.o
a - build/Debug/GNU-MacOSX/gplot_errors.o
a - build/Debug/GNU-MacOSX/gplot_errorstheory.o
a - build/Debug/GNU-MacOSX/gplot_theory.o
ranlib dist/Debug/GNU-MacOSX/libgplot.a
```

Step 3: Tell other projects about your new library

While working on another project, do the following:

Go to File->Project Properties (<MyOtherProject>) ... a new window will open up.

(a) Add the library project main directory to Build->C Compiler->General->Include Directories
(b) Add the folder where the library is located (shown above) to Build->Linker->General->Additional Library

Directories
(c) Add the actual library (libXXXX.a/dll) to Build->Linker->Libraries
(d) Click Apply and OK
(e) Clean and Build your main project

Example:

Project name = gplot_test
Library project name = plot
OS Architecture = GNU-MacOSX

Output of build command:

```
cd '/Users/brash/NetBeansProjects/gplot_test'
/usr/bin/make -f Makefile CONF=Debug
"/Applications/Xcode.app/Contents/Developer/usr/bin/
make" -f nbproject/Makefile-Debug.mk QMAKE=
SUBPROJECTS= .build-conf
"/Applications/Xcode.app/Contents/Developer/usr/bin/
make"  -f nbproject/Makefile-Debug.mk dist/Debug/GNU-
MacOSX/gplot_test
mkdir -p build/Debug/GNU-MacOSX
rm -f "build/Debug/GNU-MacOSX/main.o.d"
gcc    -c -g -I../gplot -std=c99 -MMD -MP -MF "build/
Debug/GNU-MacOSX/main.o.d" -o build/Debug/GNU-
MacOSX/main.o main.c
mkdir -p dist/Debug/GNU-MacOSX
gcc     -o dist/Debug/GNU-MacOSX/gplot_test build/
Debug/GNU-MacOSX/main.o -L../gplot/dist/Debug/GNU-
MacOSX -lgplot
```

Note that for the final gcc command, which links the executable, the added library directory is present, with the “-L” directive, and then the actual static library follows, with the “-l” directive. Libraries are of the form “libXXXX.a” or “libXXXX.dll”, and the compiler expects the “lib” and “.a/.dll” parts of the name to be removed (i.e. it just wants -lXXXX), so that is why NetBeans properly strips this part off for you.