NetBeans GBA User Guide

This document is intended as a comprehensive guide to three fundamental capabilities:

1. installing the basic NetBeans Software Development Environment on both Windows and Mac / Unix machines,
2. customizing NetBeans to allow development of C programs targeted for the Nintendo Gameboy Advanced (GBA) on both Windows and Mac / Unix machines, and
3. developing programming projects in two modes – either Debug mode in the full NetBeans environment or Release mode targeted for the GBA emulator.

1.0. NetBeans Installation Instructions

1.1. Windows XP / Windows Vista / Windows 7 Installation

1. Go to http://www.netbeans.org/ and press the **Download Free NetBeans IDE 6.9.1** button in the center of the page. Look for the C/C++ column and press the **Download** button at the bottom. Save and run the file.

2. If you see the message depicted in Figure 1 below continue with step 3. Otherwise skip to step 4.

Figure 1 - JDK Not Found Error

3. Go to http://www.java.com/ and press the **Free Java Download** button in the middle of the page. On the next page click the **Agree and Start Download** button. Once it is finished downloading, run the file and install java using the default settings. Then retry installing NetBeans.

4. You should now be seeing the NetBeans IDE Installer as shown in Figure 2 below. Install NetBeans with the default settings.

5. Once NetBeans is installed, go to http://www.cygwin.com/ and press the **Install or update now!** link in the center section of the page. Download and run the setup.exe file. Until you reach the **Select Packages** page continue through the cygwin install using the default settings. Select a download site when prompted. Click **Ok** when a window message titled Setup Alert about your first time installing Cygwin 1.7.x shows up.

6. Once you are on the **Select Packages** page shown below, make sure to select the following:

a. **gcc-core: Compiler**
b. **gcc-g++: C++ compiler**
c. **gdb: The GNU Debugger**
d. **make: the GNU version of the 'make' utility**.

Use the + button next to Devel to reveal the Packages as shown in Figure 3. Click on Skip to change it to the version number to make sure the packages will install. Once you have selected the needed packages select **Next.** On the next page select **Next** to accept installing the needed dependencies as well. Finish

installing cygwin with the default settings.

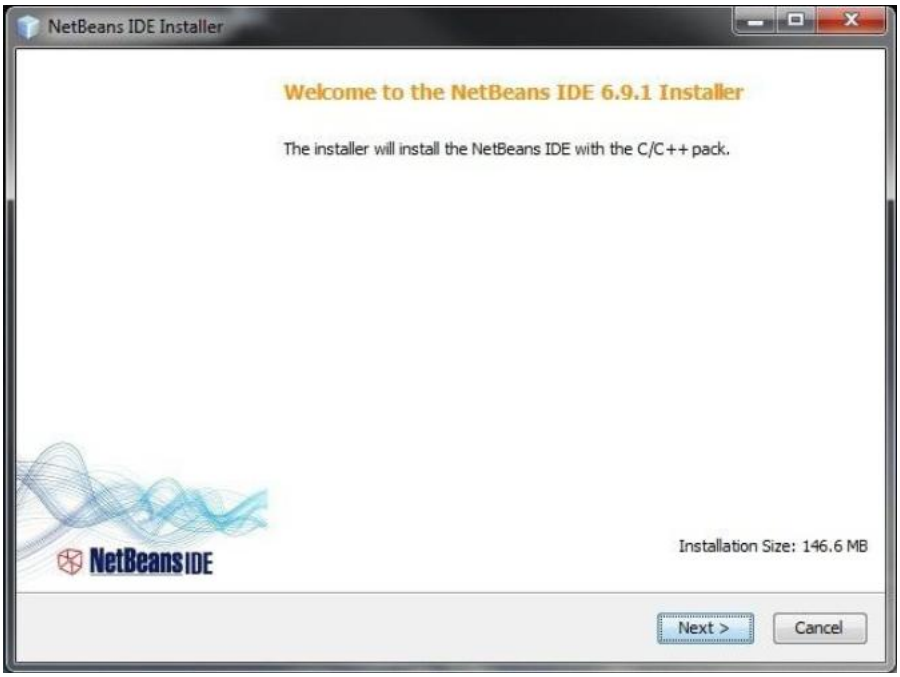


Figure 2 - NetBeans Welcome Screen

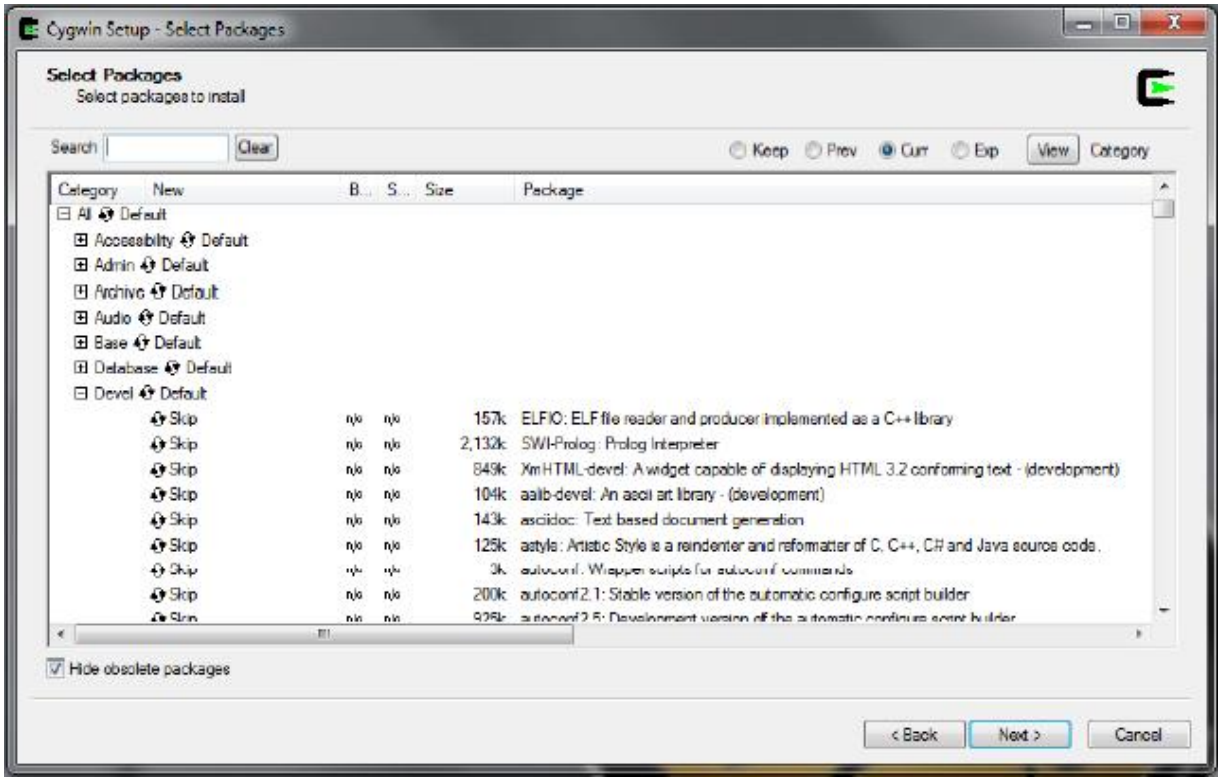


Figure 3 - Cygwin Selections

7. Once Cygwin is done installing, you need to add the compiler to your path.

a. Open the start menu, right click on My Computer and choose properties as shown in Figure 4.

Figure 4 - Opening My Computer Properties

b. If you are on **Windows XP** select the advanced tab. If you are in **Windows Vista** or **Windows 7** select the **Advanced system settings** on the left as shown in Figure 5.

Figure 5 - Windows XP

Windows Vista / Windows 7

c. Click **Environment Variables,** giving you the views in Figure 6.

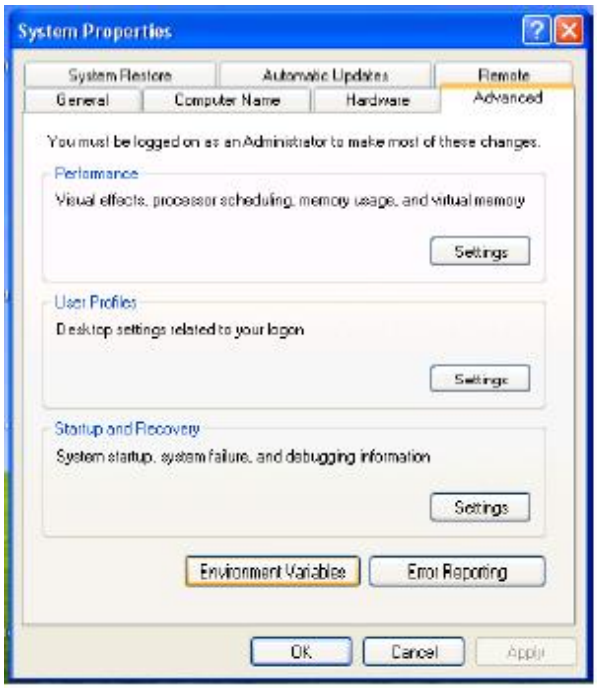
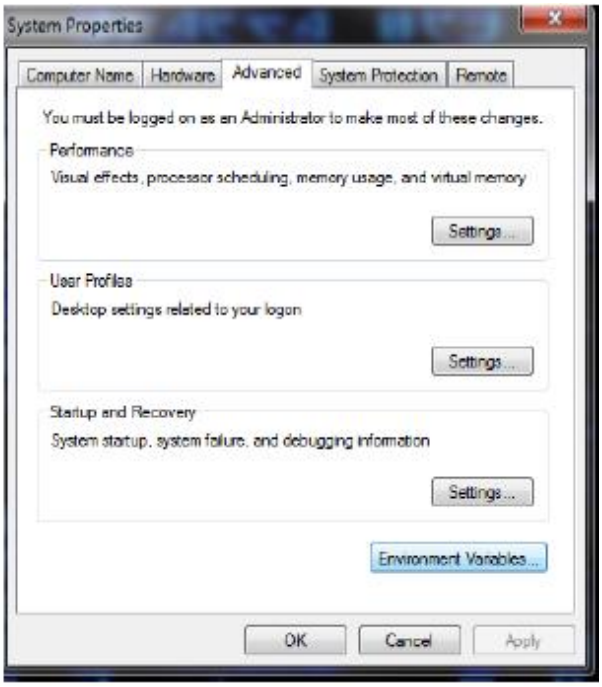


Figure 6 - Windows XP



Windows Vista / Windows 7

d. In the system variables panel select **Path** and press the **Edit…** button on the view in Figure 7 (you may need to scroll down to find Path).

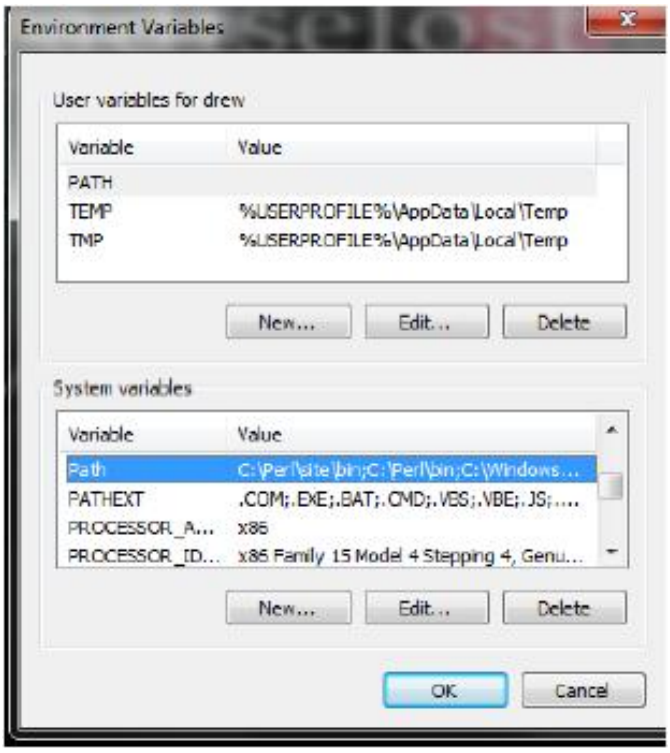


Figure 7 - Choosing the Path Property

e. At the end of the path add a '**;**' if there is not one already and '**C:\cygwin\bin**;' . Make sure there is no space between the '**;**' and the '**C**'. Click Ok on all of the open windows to accept the changes.

Figure 8 - Editing the Path Property

8. test your installation by launching NetBeans (e.g. Applications > NetBeans > NetBeans 6.9.1), creating a new C/C++ Application, with a default C file called "main.c". Make sure it has a good path, project name, and that C is selected instead of C++.
You are now finished with the basic NetBeans installation – go to the GBA Installation section.

9. If you wish, you can change the run configuration preferences to send the console output to the output window instead of a fresh pop-up window and Run your application. You should see "Hello, World!" print to the console. (See section 3.4 below)

1.2. MacOS and Linux NetBeans Installation

1. Make sure you have XCode Tools installed (either from the Snow Leopard DVD or from the Leopard Extra Tools CD), or the GNU Compiler Collection (GCC)from your Linux distribution's software tool. To check this, open a terminal (Applications > Utilities or Applications > Accessories) and type the following (what you type is **bold**, what you should see is not):

$ **make**
make: *** No targets specified and no makefile found. Stop.
$ **gcc**
i686-apple-darwin10-gcc-4.2.1: no input files.

If you see "command not found" you need to (re)install XCode Tools or GCC and Make.

2. Download NetBeans. Choose any with C/C++ support. We suggest the C/C++ specific version, but if you think you might be using it for other development, you may choose a different option. On Linux, you may have this already in your software distribution, though you may still want the newest version available.
http://netbeans.org/downloads/index.html

3. If you are on Mac, Open the downloaded .dmg and run the installer. You should see a big friendly green check mark and the message "The installation was successful."

4. test your installation by launching NetBeans (e.g. Applications > NetBeans > NetBeans 6.9.1), creating a new C/C++ Application, with a default C file called "main.c" as shown in Figure 9. Make sure it has a good path, project name, and that C is selected instead of C++. Open and edit the main file, adding a "Hello, World!" printf as in Figure 10.

5. If you wish, you can change the run configuration preferences to send the console output to the output window instead of a fresh pop-up window (Section 3.4 below) and Run your application. You should see "Hello, World!" print to the console.

You are now finished with the basic NetBeans installation – go to the GBA Installation section.

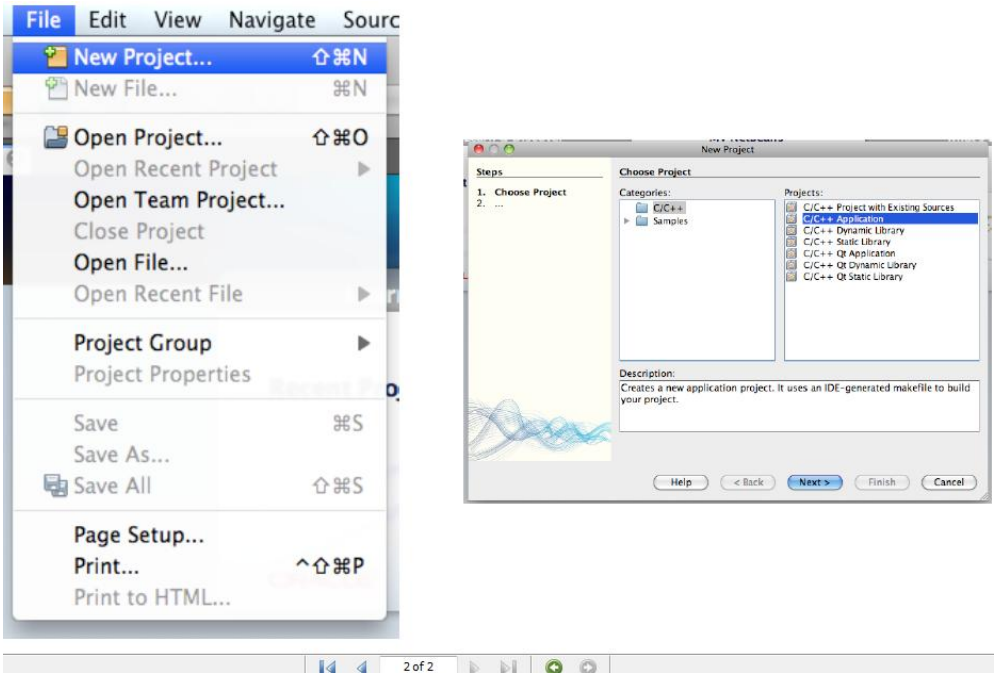


Figure 9 -Make a New Project

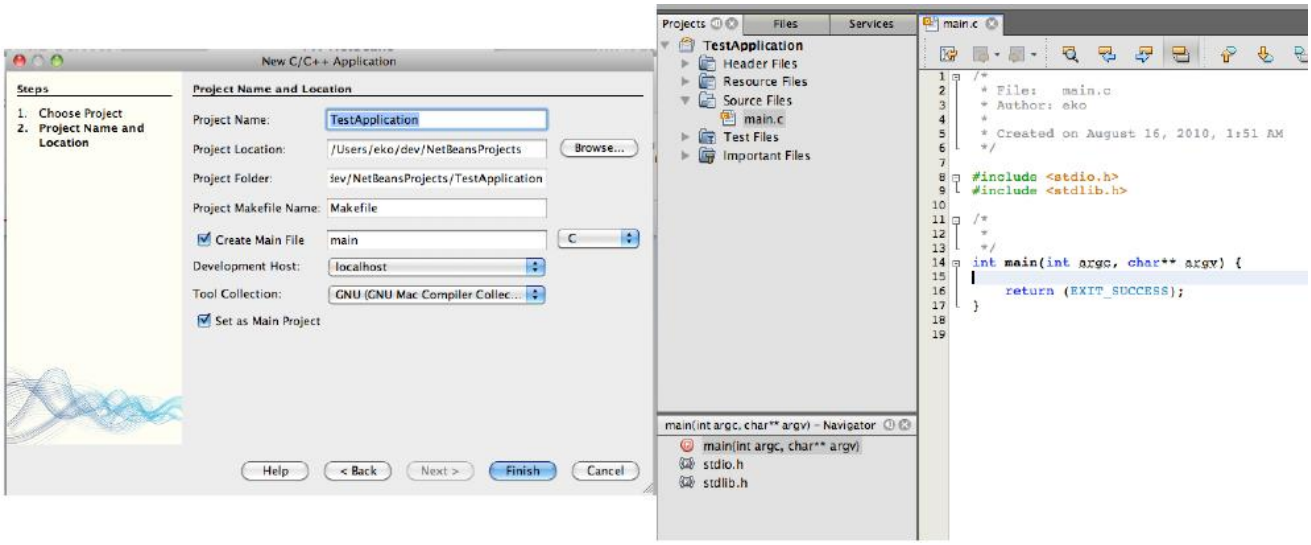


Figure 10 - Edit and Run the Project

2.0 NetBeans GBA Support Installation

The following instructions must be followed meticulously in order to configure your NetBeans installation to develop programs for our target platform, the Nintendo Game Boy Advanced (GBA). Take courage. This is a daunting series of instructions, but it only needs to be done once.

2.1 Obtaining the Tools

1. Go to T-square and download '**CommonGBAFiles.zip'** and unzip it. You will be using these files for every project you make, so make sure you put the unzipped folder somewhere convenient.

2. Also on T-square there is a folder containing necessary development tools for the Windows operating system. Mac/Unix users can download the development tools from:

http://sourceforge.net/projects/devkitpro/files/devkitARM/

Open whichever one pertains to you, and unzip that version of **devkitARM** saving the whole folder somewhere convenient. By default, Windows users could put it directly on the C: drive so that the location of the devkitARM tools is "C:\devkitARM". The default Mac/Unix location is "/opt/devkitARM".

3. Finally, you must download a Gameboy emulator. There are a few options, but in this class we'll use VisualBoyAdvance. You can download appropriate versions from the following links and save the whole collection in a safe place. By default, Windows users might use "C:\vba." Mac/Unix users could use "/usr/bin."

- Windows: http://prdownloads.sourceforge.net/vba/VisualBoyAdvance-1.7.2.zip
- Mac: http://prdownloads.sourceforge.net/vba/VisualBoyAdvance-1.7.2-MacOSX.dmg.gz
- Linux: http://sourceforge.net/projects/vba/files/VisualBoyAdvance/1.7.1/VisualBoyAdvance1.7.1-SDL-linux-glibc22.tar.gz/download

2.2 Configuring your First Project

1. Use your Finder / Explorer to determine exactly where the system utilities you just obtained are located. You must know where the following are on your hard drive. Windows users should note that NetBeans is going to want to see forward slash '/' not back slash '\' in the file paths.

- devkitARM/bin (default: "C:/devkitARM/bin" or "/opt/devkitARM/bin")
- cygwin/bin (default: "C:/cygwin/bin" or "/usr/bin")
- VisualBoyAdvance.exe (default: "C:/vba/ VisualBoyAdvance.exe" or "/usr/bin/open")

2. Go to the "common" folder you downloaded in step 1, and open "Makefile" in the text editor of your choice. Scroll down to the lines pictured below:

```
# --- Project Settings
PRODUCT_NAME       = cs1372
SOURCES            = ######## Your sources here

# --- System Settings
DKPATH             = ######## Path to DevKitARM bin directory
CCPATH             = ######## Path to gcc bin directory
VBASIM             = ######## Path to VBA (or /usr/bin/open on mac)
```

3. Replace whatever occurs at the comments (########) in the “DKPATH =” line with the path to **devkitARM** which you downloaded in step two. It should end with ‘/devkitARM/bin’. The “CCPATH =” line should have the path to your gcc compiler. In OS X or Linux this is typically ‘/usr/bin’, and on Windows it should be ‘C:/cygwin/bin’. For Windows, make sure you use forward slashes. Finally, the “VBASIM =” line should point to your Gameboy emulator (or as the image shows you can use /usr/bin/open on a mac). Example Makefiles with the paths filled out for OS X (left) and Windows (right) are shown in Figure 11. Make sure to delete the comments on each line as well. It might also be a good idea to add “main.c” to the SOURCES line. Save the Makefile.

4. Open Netbeans and make a new project. We are going to replace the contents of the main.c file and the Makefile with the common GBA files we obtained in step 1 and add the gba_wrapper.c file. The easiest way to accomplish this is to open the common GBA main.c file in an editor and copy its complete contents to your paste buffer (usually, Ctl-C). Then, open the main.c file in your NetBeans project, delete all its content and paste in the common GBA file contents. Then, expand the “Important Files” folder on the left, and change the contents of the Makefile to the contents of the common GBA Makefile in the same way.. Finally, you'll want to go to the project folder in your file viewer and **Copy** the “gba_wrapper.c” file from the “common” folder to the project folder. When you're done the project folder should look like the image in Figure 12. (Note the file path at the top if you are unsure of where projects are located).

```
# --- Project Settings
PRODUCT_NAME        = cs1372
SOURCES             = main.c

# --- System Settings
DKPATH              = /opt/devkitARM/bin
CCPATH              = /usr/bin
VBASIM              = /usr/bin/open
```

Figure 11 - Mac / Linux Makefile

```
# --- Project Settings
PRODUCT_NAME        = cs1372
SOURCES             = main.c

# --- System Settings
DKPATH              = C:/devkitARM/bin
CCPATH              = C:/cygwin/bin
VBASIM              = C:/vba/VisualBoyAdvance.exe
```

Windows Makefile

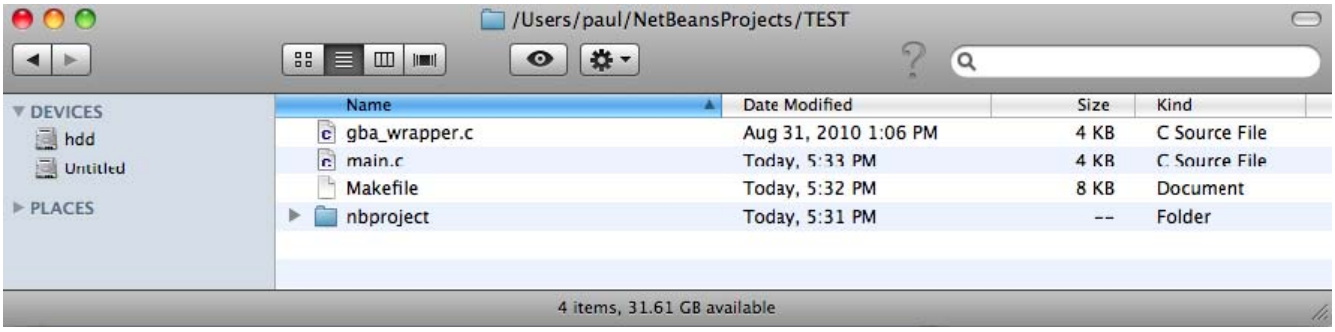


Figure 12 - The Common Files (Mac View)

5. Now go back to Netbeans. You are going to customize the Run settings by going to Run → Set Project Configuration → Customize... as shown in Figure 13. Change the Configuration box at the top to say “<All Configurations>.” Click the “Run” tab on the left and change “Console Type” to “Output Window”. Then, click on the “Linker” tab on the left and change the “Output” line to say “cs1372”. Finally, Click “OK” to save the changes. See Figure 14 for the completed configuration screen.

6. If your only source file is main.c, now you're done! If you have other source files, however, there is one more step. Expand the “Important Files” folder again, and select the Makefile that you recently copied there. Open it in Netbeans, and alter the “SOURCES =” line to reflect your new sources. For example, if your project has main.c draw.c and lib.c, then the line will look as follows: “SOURCES = main.c draw.c lib.c”.

7. Now, you should be able to run your program in Debug mode (to print on the console), by making sure the box near the top of Netbeans says “Debug” and by clicking the hammer icon to compile, and the green arrow to run. You can also run your program in the GBA emulator by switching the box to say “Release”, and clicking the hammer icon to compile, and then the green arrow to run it.

8. For future projects, rather than creating them from scratch, you can copy and rename a similar project and work from there (Section 3.3).

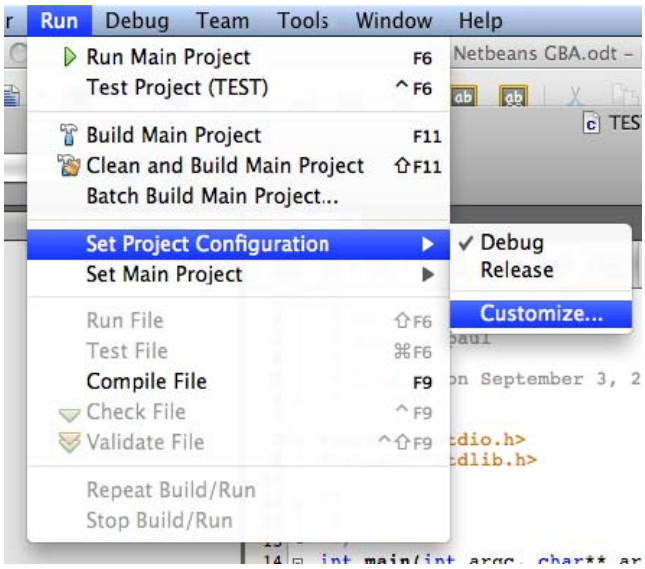


Figure 13 - Setting Project Configurations

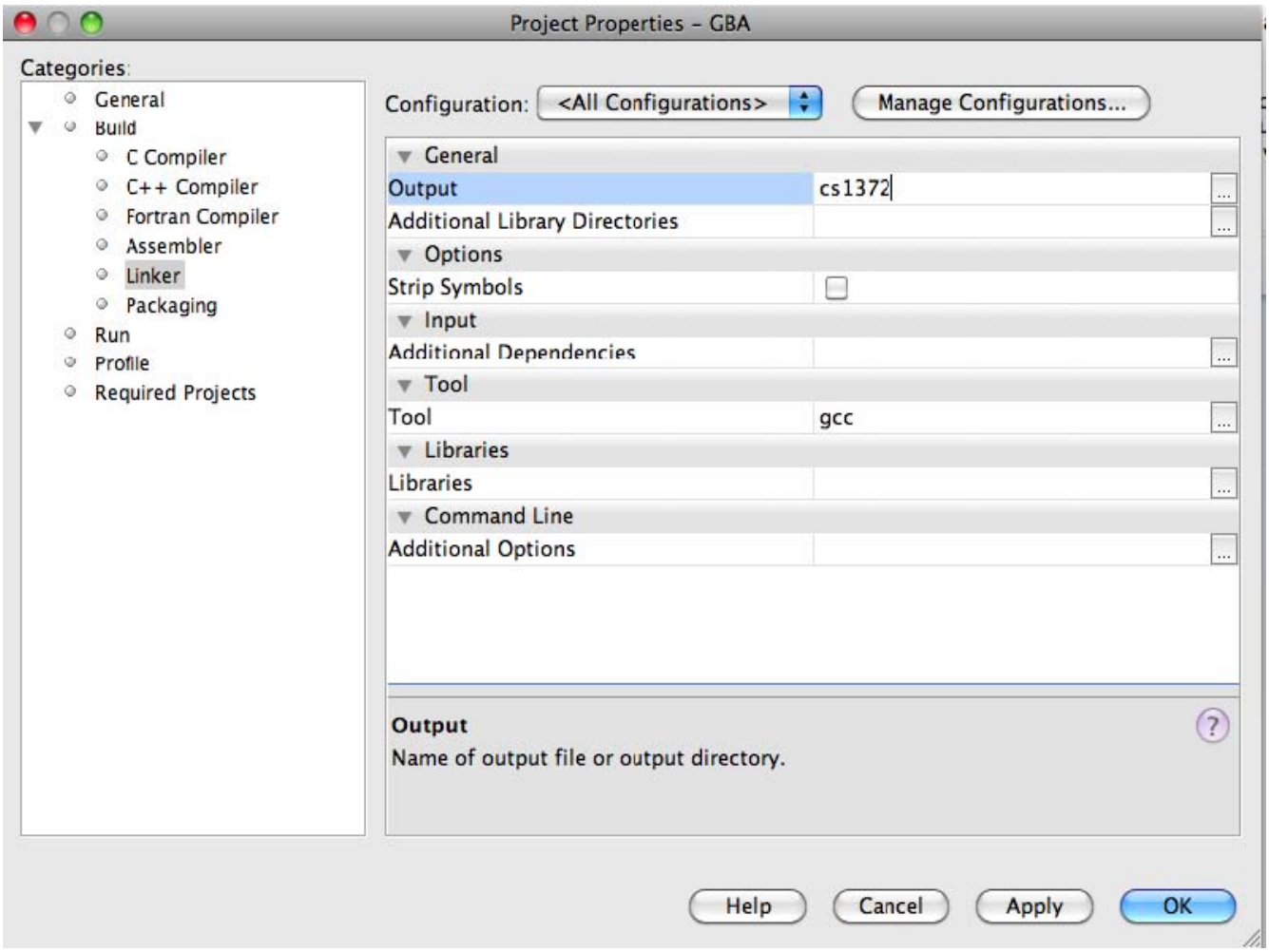


Figure 14 - Setting the Linker Configurations

3.0 NetBeans Usage Instructions

Once NetBeans (with its GBA customization if necessary) is successfully installed, follow these directions to begin developing C/C++ programs.

3.1 Testing NetBeans

To test your installation, open NetBeans. The first time it may ask you to register. Click never register or register later and continue to the start page. Once there select **File- > New Project**. Select **C/C++ -> C/C++ Application** and select **Next**. On the next page change the drop down box to **C** and click **Finish**. Type **printf("Hello, World!\n");** on line 15. Next go to **Run -> Run Main Project**. If you see it open the command prompt and print "Hello, World!" you have successfully installed everything.

3.2 Creating a New Project

1. To run a c file provided from an external source, open NetBeans and go to **File- > New Project**. Select **C/C++ -> C/C++ Application** and select **Next**. This time change the project name to **TestCode** and click finish (the file type for main should default to C). Now if you go to your **My Documents**, or **Documents** folder and then the folder **NetBeansProjects**, you should see a folder named **TestCode**. Inside the folder you should see a file called **main.c**. Fetch the code you want to run (assumed for now to be called main.cand

replace the existing main.c with this file. Now if you go back to NetBeans, and click on **Source Files -> main.c** under the project **TestCode** on the left , you should see the file you just imported. Go to **Run -> Run Main Project** to see the program run.

2. If you are using the GBA customizations and add a file, you must edit your Makefile (look in the Project under Important Files). Find the list of source files and add your new files with the ".c" extension. If you need more space on the line, a back slash '\' character at the end of the line will extend the list to the next line in the Makefile. It is important that you make no other changes to the Makefile.

3.3 Re-Using Projects

1. To copy a project in order to retain a working version and add functionality, open NetBeans and select the project to be copied in the Projects window. Right click to open the context menu and select Copy. You will be prompted for the new project name. Enter that, and check the project location – it should be the same folder as your other NetBeans projects. Accept all the defaults and run the new project.

2. To switch between projects, select the project you want to run, right click for the context menu and select "Set As Main Project".

3.4 Redirecting Output in NetBeans

This guide will help you make the text that appears in the black command window when you run a program appear in the NetBeans window instead.

1. If you have not done so already, install NetBeans using the appropriate guide under Installation Instructions in the resources section of the 1372 T-square tab.
2. Once you have NetBeans installed and set up properly, open NetBeans.
3. Click on **Run -> Set Project Configuration -> Customize**.
4. Select the **Run** category on the left.
5. On the line that says **Console Type**, change the drop down menu to **Output Window**. Press **OK** to accept the changes.
6. Now when you run a program you should see console output in the output window at the bottom of NetBeans.