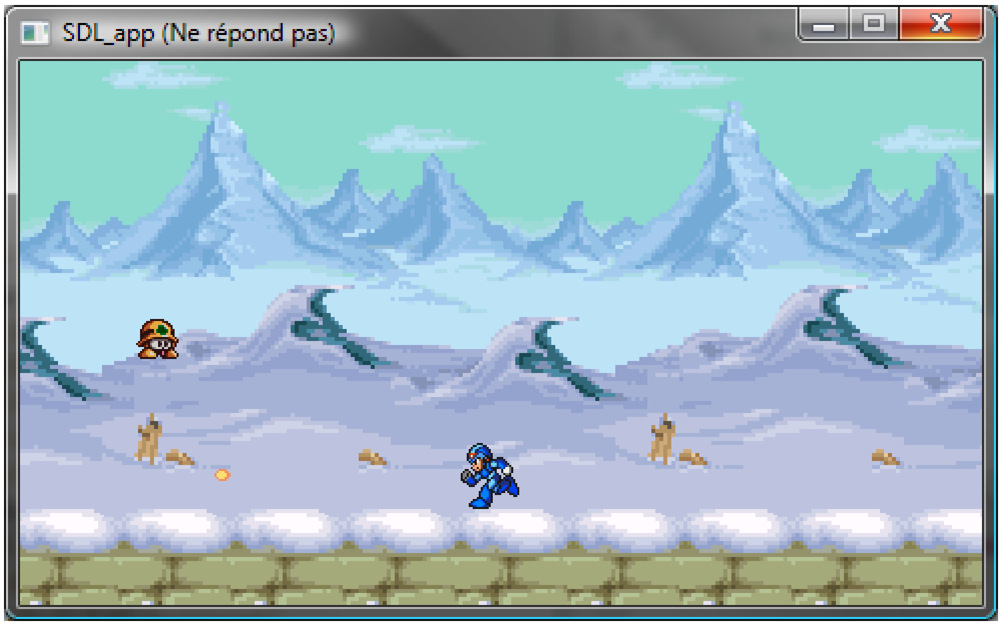
# Setup Windows – Project ArmConsole

## Contents (Click to jump to title)

## The project

This is what you should see if you follow each step of this tutorial.



## Requirements

- Java SDK/JRE (probably already install, eclipse need this to run)
- Eclipse Galileo
  - CDT (C/C++ Development Toolkit) plug-in for eclipse http://download.eclipse.org/tools/cdt/releases/galileo
  - Subclipse (optional) http://subclipse.tigris.org/update_1.6.x
- Cygwin (http://www.cygwin.com/setup.exe)
- SDL (include in the project on the SVN so do not download it now)
- TortoiseSVN (http://tortoisesvn.net/downloads)

## Installations

***IMPORTANT***
***See the troubleshooting part after all the installations.***

### Step 1 – Cygwin

Cygwin is a Unix-like environment and command-line interface for Microsoft Windows. As Windows do not have a C/C++ compiler, we need to install one and UNIX comes with GCC/G++ that are the C/C++ compiler we are looking for (there are alternatives but for the project, it is important to use Cygwin because we need a recent version of GCC/G++).

So it goes like this:

1. Run the setup.exe (http://www.cygwin.com/setup.exe).
2. Follow the installation steps (click next),
   a. I suggest you to keep the default installation folder.
   b. You need to select a mirror site in one of the steps, choose a random one.
3. Then all the available packages will be listed. Make sure you take the "gdb", "gcc", "g++" and the "make" packages in the devel category. Then hit next.
4. Go get a coffee or a beer, this may take a while...
5. After installation paste the line above (or the equivalent folder that you choose for installation) in the Environment Variables named Path (System > Advanced > Environment Variables, you should see path in the list below):

**;c:\cygwin\bin;c:\cygwin\usr\local\bin**

***IMPORTANT***
***To know if the version of gcc/g++ you have is up to date for the project, open the Cygwin Bash Window and type: gcc –v, you should see the gcc version at the bottom of the text. It must be version 4.x.x and newer.***

*Know that you can't run the programs provided by Cygwin in the Command prompt window; you'll be able to run those only in the Cygwin console window.*

I guess it's all for Cygwin.

## Step 2 – Java SDK/JRE

Just download and install the latest Java SDK and JRE if not already done.

## Step 3 – TortoiseSVN

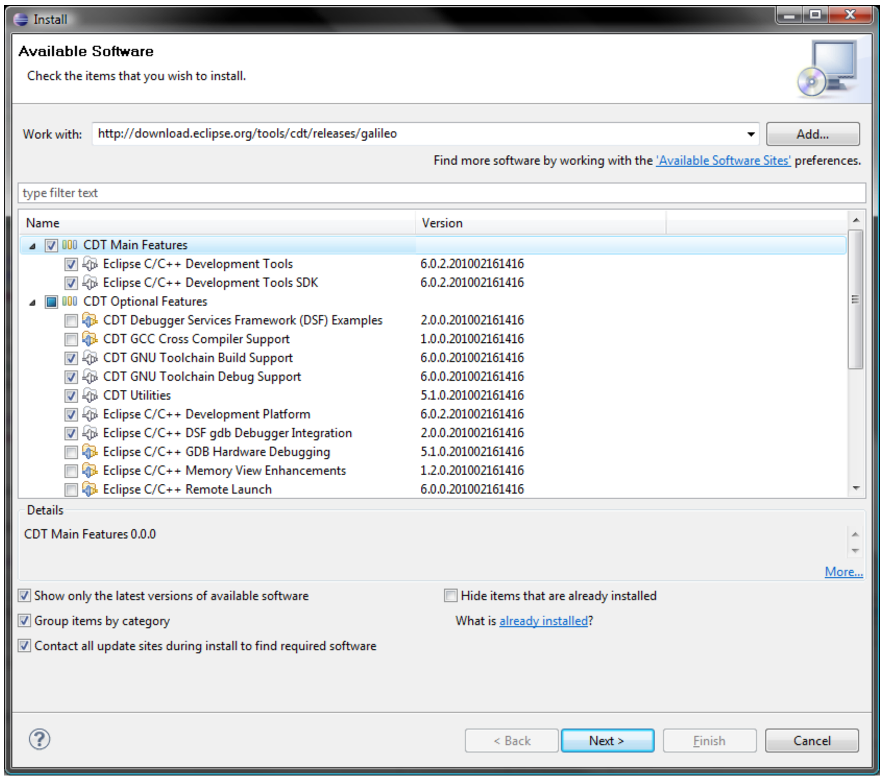
Nothing special to say, just download and install it.

## Step 4 – Eclipse

1. Unzip Eclipse somewhere in your system (eclipse is ready to use, do not need any setup). I suggest something like "C:\program files\eclipse".
2. Then open eclipse.

*I got a strange error when I opened Eclipse Galileo, saying that eclipse wasn't able to find the JVM or the javaw.exe file and exiting with the code -1. To fix this, I created a shortcut and added -vm "C:\directory to java folder\Java\jre6\bin\javaw.exe" as a parameter*

3. Go to Help > Install New Software... then in the "work with" text box, paste this: http://download.eclipse.org/tools/cdt/releases/galileo
4. Check the "CDT Main Features"


Install
Available Software
Check the items that you wish to install.

Work with: http://download.eclipse.org/tools/cdt/releases/galileo

Add...
Find more software by working with the 'Available Software Sites' preferences.
type filter text

| Name | Version |
|---|---|
| ✓ CDT Main Features | |
| ✓ Eclipse C/C++ Development Tools | 6.0.2.201002161416 |
| ✓ Eclipse C/C++ Development Tools SDK | 6.0.2.201002161416 |
| ■ CDT Optional Features | |
| ☐ CDT Debugger Services Framework (DSF) Examples | 2.0.0.201002161416 |
| ☐ CDT GCC Cross Compiler Support | 1.0.0.201002161416 |
| ✓ CDT GNU Toolchain Build Support | 6.0.0.201002161416 |
| ✓ CDT GNU Toolchain Debug Support | 6.0.0.201002161416 |
| ✓ CDT Utilities | 5.1.0.201002161416 |
| ✓ Eclipse C/C++ Development Platform | 6.0.2.201002161416 |
| ✓ Eclipse C/C++ DSF gdb Debugger Integration | 2.0.0.201002161416 |
| ☐ Eclipse C/C++ GDB Hardware Debugging | 5.1.0.201002161416 |
| ☐ Eclipse C/C++ Memory View Enhancements | 1.2.0.201002161416 |
| ☐ Eclipse C/C++ Remote Launch | 6.0.0.201002161416 |

Details
CDT Main Features 0.0.0

✓ Show only the latest versions of available software
☐ Hide items that are already installed
✓ Group items by category
What is already installed?
✓ Contact all update sites during install to find required software

*I have also the following package in the CDT Optional Features category installed: CDT GNU Cross Compiler Support, Toolchain Build Support, Utilities, Eclipse C/C++ Dev platform and DSF gdb Debugger Integration but I don't know if they're necessary for CDT to work properly.*

5. Then click next till the end (it may be necessary to check I accept the blah blah) and wait for the installation.

So now, the installation of eclipse is almost over. At that point, you should be able to make a simple HelloWorld project in C++ and compile it using the Cygwin toolchain.

## Optional – Subclipse

Subclipse allow you to do checkouts and commit stuff directly in Eclipse.

1. In Eclipse, go to Help > Install New Software... then in the "work with" text box, paste this:
   http://subclipse.tigris.org/update_1.6.x
2. Check Subclipse and follow installation steps.

So now, you'll have the "Team" option with the SVN options when you right-click. There's also a perspective named "SVN Repository Exploring" that comes with the plug-in.

# Setup – Arm Console / Megaman Demo

## Step 1 – SVN Checkout

SVN Checkout consists of copying all the project files from the SVN server to your computer, so you can work on it locally without any risk of destroying it.

1. In your Eclipse Workspace Folder, create a new Folder for the project.
2. Right-click on it and click SVN Checkout… then enter the following URL: https://armconsoledemogame.googlecode.com/svn/trunk

Now all the latest project files are in that folder.

***IMPORTANT***
***Do a SVN Update on the project folder (possible to do in Eclipse with Subclipse) BEFORE you begin to work on it and do it often while you work on it.***

## Step 2 – Eclipse project

To be fully ready to work on the project, you need to follow these 2 last steps:

- Creating the project
- Configuring the project

### Creating the project

You need to create a project in Eclipse and link it with the existing project folder. So this is how you do it:

1. File > New > C++ Project
2. Enter the Project Folder Name as the Eclipse Project name; this should automatically link the project with the existing folder. (Ignore the error that eclipse gives you)
   - If the project didn't linked with the existing Project Folder that we have created above, linked it manually: new > folder > advanced > link
3. Choose Empty Project as the project type and then uncheck the checkbox below (Show project types and toolchains only if they are supported on the platform) and Cygwin GCC should appear in the Toolchains list.
4. Choose Cygwin GCC as the toolchains (you've guessed it, I know)
5. Click next till the end and you should have a project in eclipse with all the project files downloaded!

### Configuring the Eclipse to run a project using SDL

If everything went right till you reach that step, you should have an "sdl" folder in the project tree. Eclipse can't use SDL directly; we need to configure some options.

1. Select the project folder in the Project Explorer window in Eclipse.

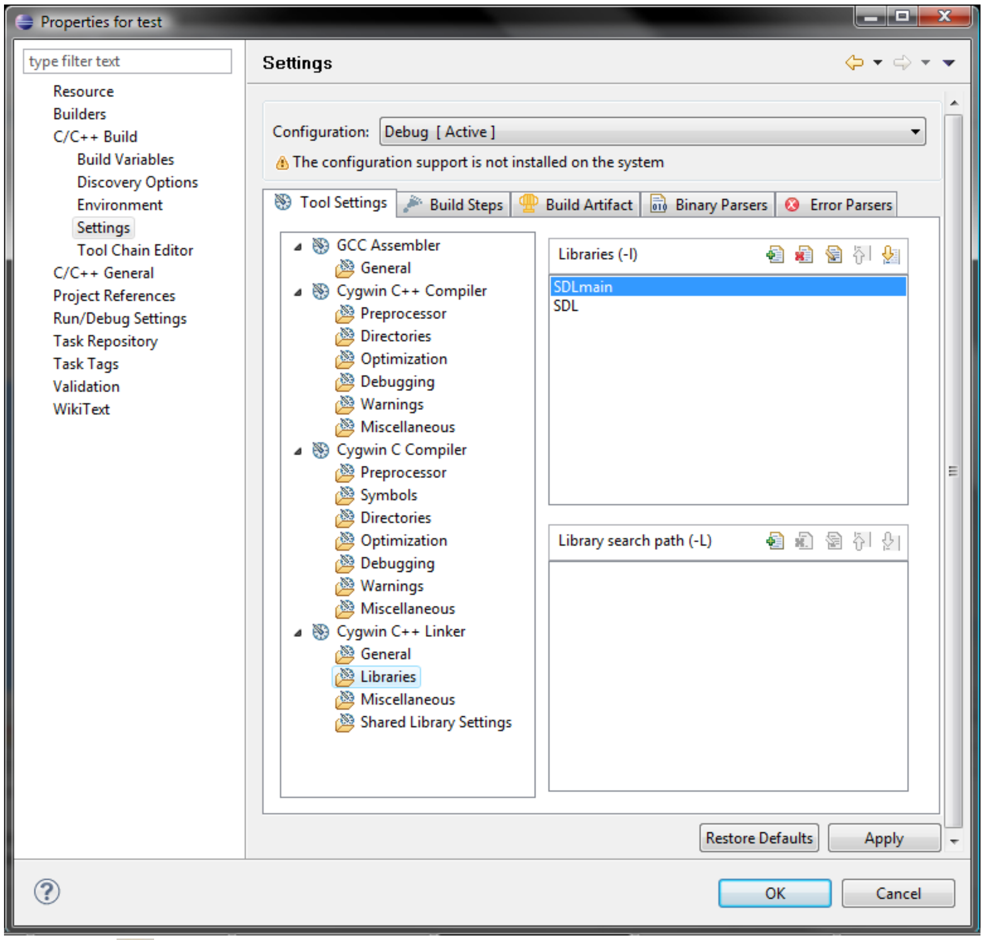
2. Project > Properties > C/C++ General, Library paths tab, then add:
   **/your project directory/sdl/lib**

*This is important, SDL won't be found if not define.*

3. Project > Properties > C/C++ Build > Environment
4. Click Add… and create a variable name Path and paste the following as a value:
   **;C:\cygwin\bin;**
5. Then click apply on the bottom right corner of that window (ignore the error that eclipse throws again).

*This is necessary to Build project; if this is not set, the "make" program won't be found.*

6. Go to Tool Chain Editor in the same window (left list) and just make sure the Current toolchain is Cygwin GCC.
7. Go to Settings and then select the Tool Settings Tab and then Cygwin C++ Linker > Libraries.



Configuration: Debug [ Active ]

The configuration support is not installed on the system

Libraries (-l)
SDLmain
SDL

8. Click add in the Libraries (-l), not in the Library Search Path and paste the followings in <u>ORDER</u>:
   **SDLmain**

**SDL**

9. Click Apply and Ok.
10. Last step, to be able to run any project using SDL, add a copy of the SDL.dll (project folder\sdl\lib) to your system32 directory (sysWOW64 if you have a 64-bit OS)

*You can put the SDL.dll file beside your project.exe in the Debug folder in your Eclipse project folder, but this won't change anything unless you have multiple SDL projects each using its own SDL version.*

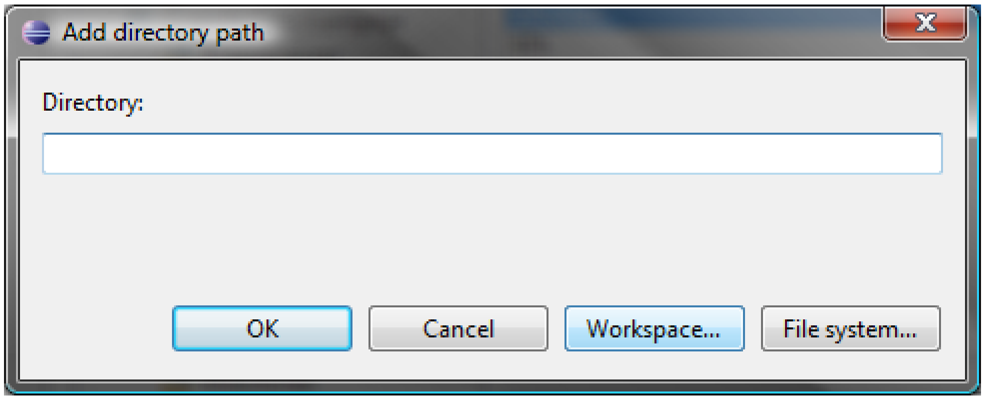
By now, you should be able to build and run a C/C++ project using SDL in eclipse.

## Configuring other SDL modules

If you want to use the other SDL modules like SDL_image, SDL_mixer (we'll need this to build and run the project) and SDL_ttf, you need to point the SDL's lib folder to make Eclipse able to see them.

1. Project > Properties > C/C++ Build > Settings and in the Cygwin C++ Linker at the bottom-right of the Libraries submenu, click the little add button beside the Library Search Path (-L). (see the picture in the last part)

You should see that panel.



2. Click Workspace... then find the lib folder (normally in projectFolder/sdl/lib)
3. Click OK.
4. Then add the followings to the Libraries (-l) list:

**SDL_mixer**

If needed (not for the project)

**SDL_image**

**SDL_ttf**

5. Last step is to add the complementary module .dll file to your system32 directory (sysWOW64 for 64-bit OS). E.g. SDL_mixer.dll

That's it for the complementary modules of SDL; you should be able now to call functions from these modules.

## Configuring the Include paths

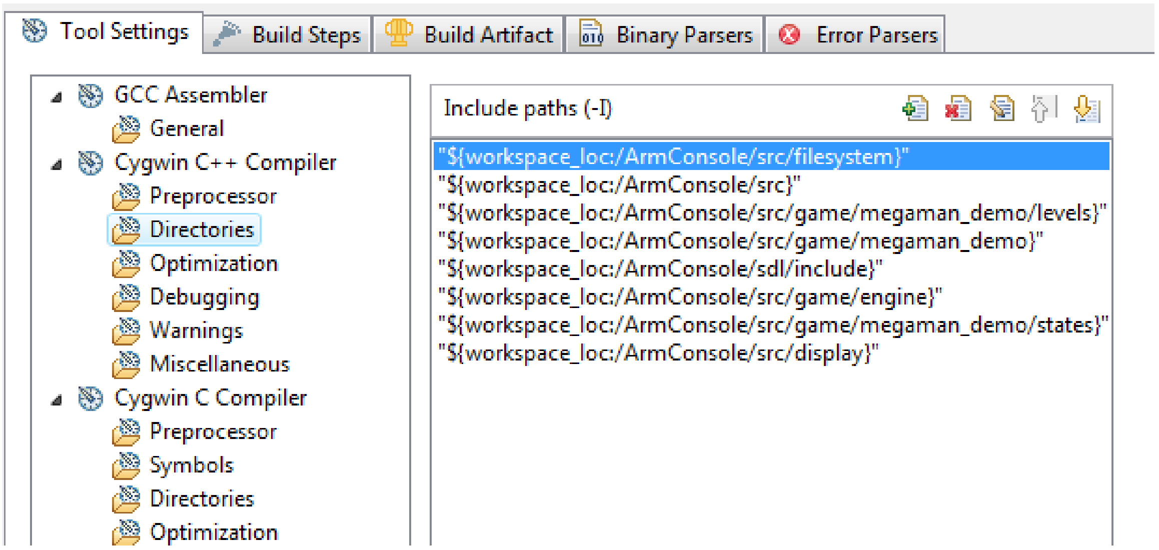
If you try to build the project now, you'll get lots of errors. It's because the compiler don't find all the .h files that are included everywhere in the project. We need to point particular directories for the compiler to be able to build.

Go to Project > Properties > C/C++ Build > Settings, choose the Directories submenu in Cygwin C++ Compiler (not C compiler) and add the following paths:

From the workspace IN ORDER (you can reorder it after):

- ✓ /src/filesystem
- ✓ /src
- ✓ /src/game/megaman_demo/levels
- ✓ /src/game/megaman_demo
- ✓ /sdl/include
- ✓ /src/game/engine
- ✓ /src/game/megaman_demo/states
- ✓ /src/display

It should look like this.



```
"${workspace_loc:/ArmConsole/src/filesystem}"
"${workspace_loc:/ArmConsole/src}"
"${workspace_loc:/ArmConsole/src/game/megaman_demo/levels}"
"${workspace_loc:/ArmConsole/src/game/megaman_demo}"
"${workspace_loc:/ArmConsole/sdl/include}"
"${workspace_loc:/ArmConsole/src/game/engine}"
"${workspace_loc:/ArmConsole/src/game/megaman_demo/states}"
"${workspace_loc:/ArmConsole/src/display}"
```

At that point, the project should compile, an executable file should be created and you should be able to run the file so (at the moment this is written) you should see something like the image at the beginning of this document.

## Troubleshooting

If you get an error like that when you are trying to build the project:

```
make all
src/testMain.d:1: *** multiple target patterns.  Stop.
```

Download a new make.exe and place it in Cygwin/bin.
http://armconsoledemogame.googlecode.com/files/make.zip

If you get warning like: unresolved inclusion, find the include file in the project and add its path to the include paths in the project properties (Cygwin C++ Compiler > Include paths)

## Links

### SVN for the projects

The project is two projects; the main part is the game engine:
http://code.google.com/p/armconsole/

The second part is the demo game that is developed on pc:
https://code.google.com/p/armconsoledemogame/

### Tutorials

Tutorial that you should read absolutely if you don't know much about SDL
http://lazyfoo.net/SDL_tutorials/index.php

### References

Useful to make SDL compile in Eclipse
http://www.lazyfoo.net/SDL_tutorials/lesson01/windows/eclipse/index.php

Useful for installing Cygwin
http://phiva.blogspot.com/2006/03/windows-c-avec-eclipse-cygwin-et-sdl.html

Informations on Cygwin
http://en.wikipedia.org/wiki/Cygwin

Installation instructions for Subclipse
http://subclipse.tigris.org/servlets/ProjectProcess;jsessionid=0369E965569282D69BEBEAEDE3A9D60E?pageID=p4wYuA