

# How to Compile and Install ORTS on Linux

By ENS Chris McIntyre

- 1) Install Linux on computer (this setup used Suse 10.1 installed with all libraries and programs)
- 2) Open a terminal window.
- 3) As the root user, make a directory under “/root” called “ORTS” (“mkdir /root/ORTS”)
- 4) Enter the directory (“cd /root/ORTS”)
- 5) Download or otherwise obtain the following files and place them in the “/root/ORTS” directory:
  - a. orts3.tgz and orts\_data.tgz  
([http://www.cs.ualberta.ca/~mburo/orts/src\\_snapshot/snap.html](http://www.cs.ualberta.ca/~mburo/orts/src_snapshot/snap.html))
  - b. boost version 1.33.1 or higher  
([http://prdownloads.sourceforge.net/boost/boost\\_1\\_33\\_1.tar.gz?download](http://prdownloads.sourceforge.net/boost/boost_1_33_1.tar.gz?download))
  - c. boost-jam version 3.1.13-1 or higher  
(<http://prdownloads.sourceforge.net/boost/boost-jam-3.1.13-1-linuxx86.tgz?download>)
  - d. SDL version 1.2.11 or higher  
(<http://www.libsdl.org/release/SDL-1.2.11-1.i386.rpm>)
  - e. SDL\_devel version 1.2.11 or higher  
(<http://www.libsdl.org/release/SDL-devel-1.2.11-1.i386.rpm>)
  - f. SDL\_net version 1.2.6-1 or higher  
([http://www.libsdl.org/projects/SDL\\_net/release/SDL\\_net-1.2.6-1.i386.rpm](http://www.libsdl.org/projects/SDL_net/release/SDL_net-1.2.6-1.i386.rpm))

- g. SDL\_net\_devel version 1.2.6-1 or higher  
([http://www.libsdl.org/projects/SDL\\_net/release/SDL\\_net-devel-1.2.6-1.i386.rpm](http://www.libsdl.org/projects/SDL_net/release/SDL_net-devel-1.2.6-1.i386.rpm))
  - h. freeglut version 2.4.0 or higher  
(<http://prdownloads.sourceforge.net/freetglut/freetglut-2.4.0.tar.gz?download>)
  - i. glew version 1.3.4 or higher  
(<http://prdownloads.sourceforge.net/glew/glew-1.3.4-src.tgz?download>)
- 6) Install the 4 SDL library files using the RPM program (“rpm -ivh \*.rpm”)
  - 7) Unzip all of the gzipped files (“gunzip \*gz”)
  - 8) Untar each tar’ed file (“tar xpf \*.tar”)
  - 9) Copy the bjam program from the “boost-jam...” directory to the “boost...” directory (“cp ./boost-jam\*/bjam ./boost\*/”)
  - 10) Enter the “boost...” directory (“cd boost\*”)
  - 11) Compile and install boost using the bjam builder (this compilation takes a while, so go get some coffee) (“bjam “-sTOOLS=gcc” install”)
  - 12) Make a “boost” subdirectory under “/root/ORTS/orts3.snapshot/libs/kernel/src”
  - 13) Copy the “boost” subdirectory into the “/root/ORTS/orts3.snapshot/libs/kernel/src/boost” directory (“cp -r ./boost/\* ../orts3.snapshot/libs/kernel/src/boost”)
  - 14) Enter the freeglut directory (“cd ../freetglut\*”)
  - 15) Configure and make and install the binaries (“./configure; make; make install”)
  - 16) Enter the glew directory (“cd ../glew\*”)
  - 17) Make and install the binaries (“make; make install”)
  - 18) Enter the orts directory (“cd ../orts3.snapshot”)
  - 19) Open up the “makefile” file in your favorite text editor

20) Add the proper library directories to the “LIB\_DIRS” line (near the top of the file) and also to the “INC\_OPTS” line (which is in the “LINUX | CYGWIN” section). For the latter, you must put “-I/DIRECTORY” and for the former, you just put the directory.

- a. for the Suse 10.1 setup add: “/usr/lib”, “/usr/include/GL”, and “/usr/local/lib”
- b. for all others, run “updatedb” (takes a while) and then “locate glew” and “locate freeglut” to determine the directories. If they are the same, good, if not, change them. You are looking for files that will generally be under the “/usr” or “/usr/local” directories in “lib” or “include” or some such directory

21) Initialize the build (“make init”)

22) Make the binaries, with the OSTYPE set (this takes a while, so get some more coffee.

Also, there will be a lot of warnings, but there should be no errors in the compilation.)

(“set OSTYPE=LINUX && make”)

23) If it works without errors, run the following to test the program (from README file):

- a. open three terminal windows (“xterm &” 3 times)
- b. window 1: “bin/orts -nplayers 2 -nobjs 20”
- c. window 2: “bin/orts -m client -act -disp”
- d. window 3: “bin/orts -m client -act -disp”
- e. This should start a 2-player game with 2x20 objects each (no attacks, random motion actions generated in the clients). Both clients should open a window with a 2d view of the playing field, which was only used for debugging the motion and collision test code.