Sheaf System Programmer's Guide

Linux Installation How-To

# Installing on Linux

These notes assume you are using a csh or tcsh shell on Linux.

# SheafSystem installation

You will need to know the full path to your installation of the SheafSystem. We'll refer to this as <ss\_install\_dir>. For instance, you might have:

<ss\_install\_dir> = /usr/local/SheafSystem-3.0.9

The <ss\_install\_dir> contains a subdirectory for the SheafSystem itself and a subdirectory for the SheafSystemTest module. We will refer to full path to the subdirectory containing the SheafSystem itself as <ss\_dir>. It is always of the form

<ss\_dir> = <ss\_install\_dir>/SheafSystem-<version>.

# Running the installer

First decide where you want the SheafSystem Programmer's Guide installed, that is, the absolute path to the root of the SheafSystem Programmer's Guide installation tree. We'll refer to that as <install\_dir> in this document. For instance, you might choose:

<install\_dir> = /home/<user>/SheafSystemProgrammersGuide

You do not have to create this directory first, but where ever you choose, make sure you have write access to it if it exists and to its parent folder if it does not. In the remainder of this document, mentally replace <install\_dir> with your chosen path.

Now, start a nice fresh terminal window without any unknown environment lurking in its history, go to where ever you downloaded the installer to, and enter:

>sh sheafsystemprogrammersguide\_install.bin

Follow the instructions given by the installer. In particular, it will ask you "Where do you want to install?". Enter your choice for <install\_dir>.

When the installer is finished installing, it will present you with a "done" button. It takes a a short time after you've pushed done to completely finish

# Configuring the examples module

Change you working directory to examples directory <install\_dir>/examples

>cd <install\_dir>/examples

The SheafSystem libraries are delivered compiled for Gnu C++ compiler or the the Intel C++ compiler. Setting your environment depends on which compiler.

## Setting your environment for the Gnu C++ compiler

For the Gnu C++ compiler, all you need to do is set the environment variable CXX for the C++ compiler and CC for the C compiler. For instance:

>setenv CXX /usr/bin/g++

>setenv CC /usr/bin/gcc

However, the SheafSystem requires g++ version 4.2.2 or later, so make sure to you have at least this version.

## Setting your environment for the Intel C++ compiler.

The Intel compiler is not completely self-sufficient, it relies on finding the Gnu C++ standard header files, and it searches your PATH environment variable to find the Gnu compiler. The SheafSystem is compatible only with g++ version 4.2.2 or later. So you have to make sure that a g++ installation with version 4.2.2 or later is in your PATH. In particular, if your Linux installation has a g++ installation in /usr/bin that is older than 4.2.2, you have to make sure the path to a 4.2.2 or later is in your PATH before /usr/bin.

Now you'll need to know where your Intel compiler installation bin directory is. A common location is:

<intel\_bin\_dir> = /opt/intel/bin

What ever <intel\_bin\_dir> is on your system, the C++ compiler is <intel\_bin\_dir>/icpc, the C compiler is <intel\_bin\_dir>/icc, and the compiler environment script is <intel\_bin\_dir>/compilervars.csh.

Set the environment variable CXX for the C++ compiler, CC for the C compiler, and source the compilervars script.:

>setenv CXX <intel\_bin\_dir>/icpc

>setenv CC <intel\_bin\_dir>/icc

>source <intel\_bin\_dir>/compilervars.csh intel64

The above asumes you are running an a 64 bit system. Remember, replace <intel\_bin\_dir> by the actual path on your system.

## Running the configuration script

Once you've set the environment for Gnu or Intel, you're ready to configure. The examples module requires CMake version 2.8.6. If your CMake is older, you will have to upgrade. If your version is new enough, run the configuration script:

>./cmboot

The script will request the absolute path to the SheafSystem, give it <ss\_dir>, as described above

The cmboot script will configure the examples module in a subdirectory <install\_dir>/examples/build. If anything goes wrong, for instance you didn't set the CC environment variable correctly, DELETE THE BUILD DIRECTORY BEFORE YOU TRY AGAIN. If cmboot runs ok, it will finish with a recommendation to change directory to build and source the set\_env\_vars.csh script. Do it.

>cd build

>source set\_env\_vars.csh

# Running the examples

Now we're still in the build directory and ready to build the examples. To build exampe1 for instance:

>make example1

To build all the examples

>make examples

The set\_env\_vars script set LD\_LIBRARY\_PATH for you, so you can just run the example without any further prep:

>./example1

Roughly speaking the examples should be run in order, since some require the output of earlier examples.