SCIRun Installation Guide

UNIX version - Updated July 7, 2001

The ***SCIRun Installation Guide*** contains instructions and information for installing and building SCIRun, it's packages and the thirdparty libraries that it requires. See the ***Technical FAQ*** for information on troubleshooting these processes.

Table of Contents

Chapter 1: Quick Start SCIRun Installation

This chapter is a quick installation checklist.

See the next chapter ***SCIRun Installation and Build Instructions*** for more detailed instructions.

Minimum System Software Requirements

SCIRun is officially compatible with two platforms: SGI and Linux.

Refer to the chapter ***Third Party Software Installation and Build Information*** for information on installing the required software listed below.

Required OS and Build Tools			**Required Thirdparty Software**			**Optional Thirdparty Software**		
Platforms	**Software**	**Version**	**Platforms**	**Software**	**Version**	**Platforms**	**Software**	**Version**
SGI	IRIX	6.5	All	Tcl	8.3.2	All	MPEGe Lib	0.3
SGI	MIPSpro compilers	7.3.1.1m	All	Tk	8.3.2	All	Berkeley MPEG Encoder	1.5b
Linux	Kernel	2.2.14	All	[incr Tcl/Tk]	3.1.0	SGI	libimage	1.0
Linux	GCC	2.95.3	All	BLT	2.4u	All	teem	1.3
All	GNU make	3.79.1	All	Xerces-C++	1.4.0			

Installing SCIRun and Packages

1. **Obtain the source trees for SCIRun and any desired packages.**

2. **Unzip the SCIRun source tree into a directory of your choosing.**

   The chosen directory will be referred to as ***INSTALL_DIR*** from now on.

3. **Unzip the packages (if any) into *INSTALL_DIR*/SCIRun/src/Packages.**

Building SCIRun and Packages

1. **Ensure that the target install machine matches the above system requirements.**

2. **Run configure from a directory of your choosing.**

   The chosen directory will be referred to as ***BUILD_DIR*** from now on.

3. **Run GNU make in the *BUILD_DIR* directory.**

Chapter 2: SCIRun Installation and Build Instructions

This chapter provides detailed instructions for installing and building SCIRun and it's packages.

Prerequisites

Use of this document requires experience with the SGI Irix operating system and/or the Linux operating system. Knowledge of file systems, programming in C/C++, building software from tarball distributions and shell scripting will be useful if not required. The installation or upgrade of core system components (such as the OS and compilers) may be required. As such, a system administrator's assistance may be warranted.

Minimum System Software Requirements

SCIRun is officially compatible with two platforms: SGI and Linux.

The following table describes the minimum system software requirements for SCIRun. Other versions may also work, but are not officially compatible.

Click on the links in the table to get information about (software column) or to download (version column) the software.

Refer to the chapter ***Third Party Software Installation and Build Information*** for information on installing the required software listed below.

Required OS and Build Tools			**Required Thirdparty Software**			**Optional Thirdparty Software**		
Platforms	**Software**	**Version**	**Platforms**	**Software**	**Version**	**Platforms**	**Software**	**Version**
SGI	IRIX	6.5	All	Tcl	8.3.2	All	MPEGe Lib	0.3
SGI	MIPSpro compilers	7.3.1.1m	All	Tk	8.3.2	All	Berkeley MPEG Encoder	1.5b
Linux	Kernel	2.2.14	All	[incr Tcl/Tk]	3.1.0	SGI	libimage	1.0
Linux	GCC	2.95.3	All	BLT	2.4u	All	teem	1.3
All	GNU make	3.79.1	All	Xerces-C++	1.4.0			

Minimum System Recommendations

	Processor(s)	**Main memory**	**Other hardware/software**
SGI	250 MHz R10K	256 MB	Texture mapping hardware and memory
Linux	500 MHz Pentium III	256 MB	NVIDIA GeForce3 graphics card, XFree86 version 4.1.0, nvidia drivers version 1.0 - 1251

Installing SCIRun and Packages

1. **Obtain the source trees for SCIRun and any desired packages.**

   If you have access to the SCIRun CVS repository (i.e. you are a student or staff member of the SCI Institute), then contact a member of the senior staff for information regarding CVS access.

   Otherwise, go to the SCIRun download page:

   http://software.sci.utah.edu/software/download.xml?dir=0.

2. **Unzip the SCIRun source tree into a directory of your choosing.**

   The chosen directory will be referred to as ***INSTALL_DIR*** from now on.

SCIRun can be installed into any directory, within a user home directory or a system application directory. There is no "make install" step in the build process, so SCIRun will be installed in the chosen directory.

For example, if you choose to install SCIRun into user Bob's home directory, then

INSTALL_DIR = /home/bob

Depending on how SCIRun was obtained, either

a) unzip and untar the SCIRun source tree:

```
cd INSTALL_DIR
gunzip SCIRun.x.x.x.tar.gz
tar xvf SCIRun.x.x.x.tar
```

or

b) checkout the SCIRun source tree from the CVS repository:

```
cd INSTALL_DIR
cvs checkout SCIRun
```

After installing, the ***INSTALL_DIR*** directory will have a sub directory named SCIRun.

3. **Unzip the packages (if any) into *INSTALL_DIR*/SCIRun/src/Packages.**

   For example, if installing the BioPSE package:

```
cd INSTALL_DIR/SCIRun/src/Packages
gunzip BioPSE.PKG.x.x.x.tar.gz
tar xvf BioPSE.PKG.x.x.x.tar
```

Building SCIRun and Packages

1. **Ensure that the target install machine matches the above system requirements.**

   See the ***Thirdparty Software Installation and Build Information*** for more information.

2. **Run configure from a directory of your choosing.**

   The chosen directory will be referred to as ***BUILD_DIR*** from now on.

An example of a typical configure command (please excuse the line wrap):

```
cd INSTALL_DIR/SCIRun/src
./configure --with-thirdparty=/usr/local/SCIRun_Thirdparty_32 \
            --enable-package="PACKNAME_1 PACKNAME_2 ..."
```

Where ***PACKNAME_X*** is the name of a SCIRun package installed above.

There are many options for configuring SCIRun with location as the chief option. SCIRun can be configured from within the ***INSTALL_DIR***/SCIRun/src directory, as is commonly done, which yields the following:

BUILD_DIR = ***INSTALL_DIR***/SCIRun/src

However, SCIRun can also be configured from locations other than the source directory which allows for having multiple builds for different platforms, or for for different ABI's (Applications Binary Interface) while still sharing a single source directory:

```
cd INSTALL_DIR/SCIRun
mkdir linux
mkdir sgi32
mkdir sgi64
cd linux
../src/configure CONFIGURE_OPTIONS
cd ../sgi32
../src/configure CONFIGURE_OPTIONS
cd ../sgi64
../src/configure --enable-64bit CONFIGURE_OPTIONS
```

In addition to the configure location, there are several options that may be invoked at configure time by supplying the configure command with arguments. The available ***CONFIGURE_OPTIONS*** can be found in the Appendix below.

3. **Run GNU make in the *BUILD_DIR* directory.**

To run GNU make, simply type "make" or "gmake", depending on how the command for GNU make is spelled on the target install machine:

```
cd BUILD_DIR
gmake
```

Just like configure, GNU make has some useful options. The "number of jobs" option, is the most commonly used option for building SCIRun :

```
gmake -j NUM_JOBS
```

Where ***NUM_JOBS*** is an integer. If the target install machine has multiple processors, setting ***NUM_JOBS*** to the number of processors available uses all the processors to build SCIRun in parallel, often significantly speeding up the time to build SCIRun.

Additional options for GNU make can be found on this website
http://www.gnu.org/manual/make/html_mono/make.html#SEC92

Running SCIRun

After a successful build, there will be an executable in the ***BUILD_DIR*** directory named "scirun". Running this will open the SCIRun network editor window and start a console (called the SCIRun console) in the shell where scirun was started from.

If the window does not appear, note any errors displayed in the SCIRun console and see the getting help section below. Of course, it might just be hiding behind other windows on the desktop!

Additional Documentation

Additional Documentation about SCIRun can be obtained on the web at:

http://software.sci.utah.edu/software/doc/index.html

Getting Help

There is a mailing list available for asking questions about using or installing SCIRun or to get help with resolving a problem. To subscribe to the mailing list, send email to:

majordomo@cs.utah.edu

with the following in the body of the message (no subject):

```
subscribe scirun-users
```

To use the mailing list, simply send your question or problem as an email to the list:

scirun-users@cs.utah.edu

To aid in problem resolution, include a verbatim (cut and paste) copy of the errors displayed (if any). Also include a short description of what you we're doing (running a network, building the tree, etc.) and which version of SCIRun you have.

Chapter 3: Thirdparty Software Installation and Build Information

This chapter may help expedite installation of the required third party software. However, this information is not intended to replace the installation instructions that come with the software, but to supplement them. Please read the install instructions for each software distribution before attempting to use this information.

Prerequisites

Use of this document requires experience with the SGI Irix operating system and/or the Linux operating system. Knowledge of file systems, programming in C/C++, building software from tarball distributions and shell scripting will be useful, if not required. The installation or upgrade of core system components (such as the OS and compilers) may be required. As such, a system administrator's assistance may be warranted.

This information does not apply to the installation and upgrade of the OS and build tools, as those topics are quite involved.

Automatic Installation of All Thirdparty Software

Because of many bugs and "gotchas" in the thirdparty installation process, the easiest and most dependable means of installation is the SCI Institute's thirdparty install script.

However, users may conduct a manual installation by following the instructions on this page starting at Manual Installation of Required Thirdparty Software.

The automatic install script comes with all the necessary thirdparty software, and does not require additional downloads.

To use the automatic install script do the following:

1. **Download the install script tar ball from this web page:**

   http://software.sci.utah.edu/software/download.xml?dir=0.

2. **Put the tar ball into a directory of your choosing.**

   Unzip and untar the tar ball:

   ```
   gunzip Thirdparty_install.x.x.x.tar.gz
   tar xvf Thirdparty_install.x.x.x.tar
   cd Thirdparty_install.x.x.x
   ```

3. **Run the install script by typing:**

   ```
   python install 32
   ```

   For 32 bit binaries and libraries, or

   ```
   python install 64
   ```

for 64 bit binaries and libraries. 64 bit is available for SGI only.

You can also specify a faster parallel build by giving an integer which provides the "number of jobs" argument for GNU make For example,to build 32 bit binaries and libraries using 12 jobs, type the following:

```
python install 32 12
```

The install script will create a directory named SCIRun_Thirdparty_32 in the directory chosen in step 2 above. When it finishes, it will display a message regarding the --with-thirdparty SCIRun configure option. Cut and paste that option when running the configure command for SCIRun (see step 2 of Building SCIRun and Packages).

The install script requires Python version 1.5 or greater

Manual Installation of Required Thirdparty Software

Recommended Build Tools

You should use the same build tools to build the third party software as is required for building SCIRun itself.

Recommended Installation Location

SCIRun uses the latest versions of all the software that it requires, and can be configured to use those versions regardless of where that software is installed, whether it is all installed in the same location or not. However, if the target install machine requires older versions of the same software (for other programs) then it is recommended that all the required third party software be installed into a centralized and out -of-the-way directory rather than in a directory already occupied by older versions of the third party software.

That said, if the target install machine only needs the latest versions of software, or if it doesn't matter one way or the other, then it is perfectly safe to install the required thirdparty software into any directory including directories where older versions are already installed.

An example of a centralized and out -of-the-way location would be: /usr/local/SCIRun_Thirdparty_32 such that after installation the contents of that directory would look like this (assuming that the tarballs were unzipped and untarred in the /src directory inside /usr/local/SCIRun_Thirdparty_32):

```
/usr/local/SCIRun_Thirdparty_32/bin
/usr/local/SCIRun_Thirdparty_32/include
/usr/local/SCIRun_Thirdparty_32/lib
/usr/local/SCIRun_Thirdparty_32/man
/usr/local/SCIRun_Thirdparty_32/src/blt2.4u
/usr/local/SCIRun_Thirdparty_32/src/itcl3.1.0
/usr/local/SCIRun_Thirdparty_32/src/tcl8.3.2
/usr/local/SCIRun_Thirdparty_32/src/tk8.3.2
/usr/local/SCIRun_Thirdparty_32/src/xerces -c-src1_4_0
```

Recommended Configure Options

NOTE: ***INST_DIR*** represents the chosen installation directory for the indicated software distribution. ***INST_DIR*** would be "/usr/local/SCIRun_Thirdparty_32" for the example given above. Note that the ***INST_DIR*** does not have to be the same for each distribution, but can be if it is desired as is demonstrated in the example.

For Tcl, Tk and [incr Tcl/Tk]

```
configure --prefix=INST_DIR
```

For BLT

On SGI machines:

```
configure --with-cc=cc --prefix=INST_DIR
```

On Linux machines:

```
configure --prefix=INST_DIR
```

For Xerces-C++

First, set the indicated environment variables listed below, then type:

```
configure --prefix=INST_DIR
```

On SGI machines:

```
XERCESCROOT = XERCES_C_ROOT
TRANSCODER = NATIVE
MESSAGELOADER = INMEM
NETACCESSOR = FileOnly
CC = cc
CXX = CC
LDFLAGS = " -n32 "
LIBS = " -n32 "
CXXFLAGS = "-O -DXML_USE_NATIVE_TRANSCODER  -DXML_USE_INMEM_MESSAGELOADER \
            -n32 -DAPP_NO_THREADS"
CFLAGS = "-O -DXML_USE_NATIVE_TRANSCODER  -DXML_USE_INMEM_MESSAGELOADER \
          -n32 -DAPP_NO_THREADS"
```

On Linux machines:

```
XERCESCROOT = XERCES_C_ROOT
TRANSCODER = NATIVE
MESSAGELOADER = INMEM
NETACCESSOR = FileOnly
CC = gcc
CXX = g++
LIBS = " -lpthread "
CXXFLAGS = "-O -DXML_USE_NATIVE_TRANSCODER  -DXML_USE_INMEM_MESSAGELOADER"
CFLAGS = "-O -DXML_USE_NATIVE_TRANSCODER  -DXML_USE_INMEM_MESSAGELOADER"
```

Where ***XERCES_C_ROOT*** is the root of the unzipped and untarred xerces 1.4.0 distribution. For the example given above, ***XERCES_C_ROOT*** would be "/usr/local/SCIRun_Thirdparty_32/src/xerces -c-src1_4_0".

Chapter 4: Appendix

The appendix holds tables and details pertinent to building and installing SCIRun.

Configure Options

The following table contains descriptions of the options available for configuring SCIRun.

--enable-package	Compile the listed packages. SCIRun modules are compiled automatically. To also build the BioPSE and Uintah modules, use: --enable-package="BioPSE Uintah".
--enable-exename	Overrides the default executable name for SCIRun. The default is scirun.
--enable-threads	Threads are always enabled, but this allows one to use a different implementation. On the SGI, --enable-threads=pthreads and --enable-threads=sproc (default) are supported. On other platforms, only --enable-threads=pthreads is supported.
--enable-debug	Turn on debugging. To simply use " -g", just use --enable-debug. To use flags other than " -g", use --enable-debug="-myflags".
--enable-optimize	Turn on optimization. To simply use " -O2", just use --enable-optimize. To use flags other than " -O2", use --enable-optimize="-myflags".
--enable-largesos	Instead of building a shared object per subdirectory, build a few "large" shared objects for the top level subdirectories. This is more appropriate for distribution, and --disable-largesos (the default) is more appropriate for development.
--enable-64bit	Compile in 64 bit mode. Currently supported only on the SGI.
--enable-assertion-level=N	Set the level of assertions to the level N. 0 means no assertions are caught, and level 3 (the default) performs extensive checking (including array bounds checks).
--enable-parallel	Enable the "distributed memory" parallel SCIRun. This requires the globus library (see the next flag).
--with-globus=GLOBUS_DIR	Uses the globus libraries in GLOBUS_DIR.
--with-thirdparty=INST_DIR	INST_DIR = the directory containing the /bin, /lib, /include and directories for all the required thirdparty software.
--with-tcl=INST_DIR	INST_DIR = the directory containing the /bin, /lib, and /include directories for the Tcl distribution.
--with-tk=INST_DIR	INST_DIR = the directory containing the /bin, /lib, and /include directories for the Tk distribution. defaults to value given by --with-tcl.
--with-itcl=INST_DIR	INST_DIR = the directory containing the /bin, /lib, and /include directories for the [incr Tcl/Tk] distributions. defaults to value given by --with-tcl.
--with-blt=INST_DIR	INST_DIR = the directory containing the /bin, /lib, and /include directories for the BLT distribution. defaults to value given by --with-tcl.
--with-xerces=INST_DIR	INST_DIR = the directory containing the /bin, /lib, and /include directories for the Xerces -C distribution.
--with-mpeg [=INST_DIR]	INST_DIR = an optional argument for specifying the directory containing the /lib and /include directories for the mpeg_encode distribution. If an INST_DIR is not specified, then the default SCIRun_Thirdparty directory will be assumed. If the required header and library files are not found, or if this option is not specified, the Viewer rendering module will not have support for saving out MPEG movies.
--with-nrrd=INST_DIR	INST_DIR = the directory containing the /lib and /include directories for Gordon Kindlmann's Nrrd (Nearly Raw Raster Data) distribution. This optional flag is only valid if the Nrrd pacakge has been enabled (using --enable-package). If it has, this flag is only necessary to specify a location other than the default SCIRun_Thirdparty directory. The Nrrd package contains modules for converting between SCIRun Fields and Nrrds, for reading and writing Nrrds, and wrappers for invoking Nrrd raster processing algorithms.