SPIR115: U of S Physics Dept. Installation Instructions for GEANT4 on Linux

Ward Andrew Wurtz

28th August 2003

Contents

1 Introduction

GEANT4 is a program for tracking particles as they enter and exit geometries build from various materials. This guide assumes the reader understands the motivation for using particle simulation software and is only meant as an install and quick-start guide to GEANT4. This guide is accurate as of August 2003.

This guide assumes an intermediate knowledge of Linux. The designers of GEANT4 require the installer to be familiar with the concepts of static vs. dynamic libraries. Also, knowledge of the gcc compiler suite is recommended.

2 GEANT3 vs. GEANT4

The largest difference between GEANT3 and GEANT4 is the programming language. GEANT3 is written in FORTRAN using sequential programming techniques. GEANT4 is written in C++ using object-oriented programming techniques. GEANT4 uses the advanced concepts of C++ such as virtual functions to give the user more power and more flexibility in writing GEANT4 applications. However, it comes at a cost of increased complexity.

Also, at the time of writing this document, GEANT4 has not yet standardized some important features like histograms. This issue will be discussed at length in section 7.

Reasons for upgrading from GEANT3 to GEANT4 include:

1. GEANT4 will be updated in the future whereas GEANT3 will no longer be developed.

2. C++ is taught to students whereas FORTRAN is no longer taught.

Reasons for not upgrading to GEANT4:

1. Problems with histograms.

2. Complexity of object-oriented design.

3. More likely to contain bugs since it is newer and still being developed.

4. GEANT3 works and is well understood.

3 Systems and Compilers Used

GEANT4 was compiled and installed on three systems. These systems are described in table 1. Notice that using gcc 2.96 is not recommended, even though it is the default compiler for RedHat 7.

System	Linux 2.4.20	Linux 2.4.18	Linux 2.4.21
Architecture	Pentium 4	Pentium 3 SMP	SPARC sun4m
Distribution	RedHat 7.2	Debian 3.0	Debian 3.0
Compiler	gcc 3.0.2	gcc 3.0.4	gcc 2.95.4
Compiled GEANT4	yes	yes	yes
Tested OpenGL	yes	yes	no
Tested JAIDA	yes	yes	no
Tested DAWN	yes	yes	no

Table 1: Systems used to test GEANT4

GEANT4 is compiled with the flag -pedantic and therefore will generate many harmless warnings in the compile process. A very common warning comes from Linux header files using C pre-processor commands that are not ANSI C, such as #include_next.

4 Installing CLHEP

Before installing GEANT4, CLHEP must be installed. The version of CLHEP used is 1.8.0. To get the CLHEP source, go to reference [CLHEP] and click on ‘Installation Guide’. Choose the newest source package and download it to a temporary directory. Notice that many instructions on the Internet refer to gmake. By default, Linux uses GNU Make, unlike systems such as NetBSD. Either simlink gmake to make or use make instead. The recommendation is to put a gmake simlink in $HOME/bin.

To install CLHEP in the home directory:

```
tar -zxvf clhep-1.8.0.0.tgz
cd CLHEP
./configure
make install prefix=$HOME
```

CLHEP has now been installed in $HOME/lib. You may wish to simlink libCLHEP-g++.1.8.0.0.a to libCLHEP.a using the command

```
cd $HOME/lib
ln -s libCLHEP-g++.1.8.0.0.a libCLHEP.a
```

This last step will make life easier when installing GEANT4.

5 Installing GEANT4

5.1 Getting GEANT4

To get the source of GEANT4, goto the GEANT4 web-site [GEANT4]. Click on ‘Download’ and choose the GNU Tar source file. The current GEANT4 version is 5.2 so the file is geant4.5.2.gtar.gz. The tar-file can be untared by typing

```
cd $HOME
mkdir geant4
cd geant4
tar -zxvf geant4.5.2.gtar.gz
```

The GEANT4 libraries are unlike normal Unix libraries. The user will need read-write permissions to the directories containing the libraries for the default makefile structure to work when compiling GEANT4 applications. It's very windows-like in nature. The recommended directory structure is to put GEANT4 in $HOME/geant4/geant4.5.2/ so that other things can be put in $HOME/geant4/.

5.2 Configuring and Compiling GEANT4

Before attempting to configure GEANT4, it is highly recommended that the reader review the information on the GEANT4 web-site.

Once in the GEANT4 directory, type

```
./Configure -install
```

The program will output:

```
-- Geant4 Toolkit Installation --

This installation shell script will examine your system and ask you questions
to determine how the Geant4 Toolkit should be installed.  If you get stuck
on a question, you may use a !  shell escape to start a subshell or execute
a command.  Many of the questions will have default answers in square brackets;
typing carriage return will set the default.

On AFS it is allowed to specify either absolute or relative paths (i.e.
starting with the  username construct).

[Type carriage return to continue]
```

Read the message and press ENTER to continue. More information will be displayed. Press ENTER.

```
Much effort has been spent to ensure that this shell script will run on
any Unix system.  If despite that you can't run Configure for some reason,
you'll have to set the proper environment variables by hand and follow the
"manual" installation as specified in the Geant4 Installation Guide.

[Type carriage return to continue]
```

There should be no problems running the GEANT4 configure script on a Linux system. It has run flawlessly on both RedHat and Debian. Press ENTER to continue.

```
Definition of G4SYSTEM variable is Linux-g++.  That stands for:

1) OS : Linux

2) Compiler :  g++

To modify default settings, select number above (e.g.  2)
[Press [Enter] for default settings]
```

Since this document assumes that you have installed Linux, press ENTER.

```
I can set things up so that your shell scripts and binaries are more portable,
at what may be a noticeable cost in performance.  In particular, if you
ask to be portable, the following happens:

1) Shell scripts will rely on the PATH variable rather than using the paths
derived above.  2)  username interpretations will be done at run time rather
than by Configure.

Do you expect to run these scripts and binaries on multiple machines?  [n]
```

Select no at this stage.

```
Where is Geant4 installed?  [/home/ward/geant4]
```

The GEANT4 sources must be in the same directory as the intended destination of the GEANT4 libraries or the installation procedure will not work. So, if the sources are in /home/ward/geant4/geant4.5.2, then that is where GEANT4 should be installed. Notice that one can use the environment variable $PWD here. Type in $PWD then ENTER. The script will ask to confirm the location as it substitutes in the value of $PWD. Press ENTER to confirm the location of the GEANT4 files.

```
Do you want to copy all Geant4 headers in one directory?  [n]
```

‘Yes’ is recommended. It makes searching for header files much easier if they are all in the same place. Press y then ENTER.

```
Where will be directory to copy all Geant4 headers?
[/home/ward/geant4/geant4.5.2/include]
```

For simplicity sake, pressing ENTER is recommended.

```
(OPTIONAL) You can now customize installation directories:
G4TMP - tmp/ containing temporary dependency (.d) and object (.o) files;
G4LIB - lib/ containing final static (.a) or shared (.so) libraries;


1) G4TMP: /home/ward/geant4/geant4.5.2/tmp

2) G4LIB: /home/ward/geant4/geant4.5.2/lib

To modify default settings, select number above (e.g.  2)
[Press [Enter] for default settings]
```

Accepting the default settings is recommended. Press ENTER.

```
Please, specify directories where the Geant4 data is installed:

1) G4LEVELGAMMADATA: /home/ward/geant4/geant4.5.2/data/PhotonEvaporation

2) G4RADIOACTIVEDATA: /home/ward/geant4/geant4.5.2/data/RadiativeDecay
```

```
Data which is not part of the default Geant4 distribution:

3) G4LEDATA: /home/ward/geant4/geant4.5.2/../G4EMLOW1.1

4) NeutronHPCrossSections:  /home/ward/geant4/geant4.5.2/../G4NDL3.7

To modify default settings, select number above (e.g.  2)
[Press [Enter] for default settings]
```

Change these options if needed. Otherwise, press ENTER.

```
Please, specify where CLHEP is installed:

CLHEP_BASE_DIR: /usr/local

According to it will be set:

CLHEP_INCLUDE_DIR
CLHEP_LIB_DIR
CLHEP_LIB

You will be asked about customizing these next.
[/usr/local]
```

Use $HOME or whatever directory CLHEP was installed in. Notice that $HOME is needed, not $HOME/lib.

```
You can customize paths of you CLHEP installation:

1) CLHEP_INCLUDE_DIR: /home/ward/include
2) CLHEP_LIB_DIR: /home/ward/lib
3) CLHEP_LIB: CLHEP

To modify default settings, select number above (e.g.  2)
[Press [Enter] for default settings]
```

If these settings are correct, press ENTER. If they are not correct, modify them as given by the directions above. Notice that this will only work if libCLHEP-g++-1.8.0.0.a was simlinked to libCLHEP.a. If the library was not simlinked, modify CLHEP_LIB so that it points to the correct library.

```
Do you want to build ’global’ compound libraries?  [n]
```

It is safe to press ENTER for ‘no’ and the manual assumes that compound libraries are not used. This option should be explored in the future.

```
Do you want to compile libraries in DEBUG mode (-g)?  [n]
```

There is no recommendation for this option. Saying ‘yes’ will drastically increase the size of the installation but increases debugging power for those proficient with gdb. For this manual, this option was enabled.

```
By default ’static’ (.a) libraries are built.

Do you want to build ’shared’ (.so) libraries?  [n]
```

Select ‘yes’ here or the GEANT4 applications will balloon in size.

```
You selected to build 'shared' (.so) libraries.
Do you want to build 'static' (.a) libraries too?  [n]
```

There’s no real reason to build static libraries if shared libraries are being built. Press ENTER for the default ‘no’.

```
G4UI_NONE

If this variable is set, no UI sessions nor any UI libraries are built.
This can be useful when running a pure batch job or in a user framework
having its own UI system.
Do you want to set this variable ?  [n]
```

Press ENTER for the default ‘no’.

```
G4UI_BUILD_XAW_SESSION
G4UI_USE_XAW

Specifies to include and use the XAW interfaces in the application to be
built.  The XAW (X11 Athena Widget set) extensions are required to activate
and build this driver.  [n]
```

Since GUI stuff must be compiled in and can not be activated later, it is recommended that one select ‘yes’ for this option.

```
G4UI_BUILD_XM_SESSION
G4UI_USE_XM

Specifies to include and use the XM Motif based user interfaces.  The XM
Motif extensions are required to activate and build this driver.  [n]
```

Again, it is recommended that ‘yes’ be selected here.

```
G4UI_BUILD_WIN32_SESSION
G4UI_USE_WIN32

Specifies to include and use the terminal interface for Windows systems.
[n]
```

Select ‘no’ here (duh).

```
G4UI_BUILD_WO_SESSION
G4UI_USE_WO

Specifies to include in kernel library and use the OPACS WO terminal interface
for the OPACS system.  The OPACS tool must be installed to activate and
build this driver !  [n]
```

If you know what this option does, or why it should be used, add it to this manual. Otherwise press ENTER for ‘no’.

```
G4VIS_NONE

If this variable is set, no visualization drivers will be built or used.
Do you want to set this variable ?  [n]
```

Press ENTER for ‘no’ or be unhappy.

```
G4VIS_BUILD_OPENGLX_DRIVER
G4VIS_USE_OPENGLX

It is an interface to the de facto standard 3D graphics library, OpenGL.
It is well suited for real-time fast visualization and prototyping.  The
X11 version of the OpenGL libraries is required.  [n]
```

Select ‘yes’ here. OpenGL is required for visualization as you would expect.

```
G4VIS_BUILD_OPENGLXM_DRIVER
G4VIS_USE_OPENGLXM

It is an interface to the de facto standard 3D graphics library, OpenGL.
It is well suited for real-time fast visualization and prototyping.  The
X11 version of the OpenGL libraries and the Motif Xm extension is required.
[n]
```

Again, select ‘yes’ here.

```
G4VIS_BUILD_OPENGLWIN32_DRIVER
G4VIS_USE_OPENGLWIN32

It is an interface to the de facto standard 3D graphics library, OpenGL.
It is well suited for real-time fast visualization and prototyping.  The
Windows version of the OpenGL libraries is required.  [n]
```

Press ENTER for ‘no’.

```
G4VIS_BUILD_DAWN_DRIVER
G4VIS_USE_DAWN

DAWN drivers are interfaces to the Fukui Renderer DAWN. DAWN is a vectorized
3D PostScript processor suited to prepare technical high quality outputs
for presentation and/or documentation.  [n]
```

Since postscript rendering requires DAWN, select ‘yes’ here.

```
G4VIS_BUILD_OIX_DRIVER
G4VIS_USE_OIX

This driver is based on the "Hepvis" class library.  The OpenInventor driver
and the Hepvis class library are based on OpenInventor technology for scientific
visualization.  The X11 version of OpenInventor is required.  [n]
```

Select ‘no’. If you know what this does and why it should be used, please edit this manual. Also select no for the next question concerning OpenInventor and Windows.

```
G4VIS_BUILD_OPACS_DRIVER
G4VIS_USE_OPACS

OPACS is a visualization environment based on X windows and OpenGL. It is
written in ANSI C and is highly portable (UNIX, NT/X11, VMS). The OPACS
tool is required to activate and build this driver.  [n]
```

Select ‘no’. If you know what this does and why it should be used, please edit this manual.

```
G4VIS_BUILD_VRML_DRIVER
G4VIS_USE_VRML

These driver generate VRML files, which describe 3D scenes to be visualized
with a proper VRML viewer.  [n]
```

Select ‘no’. If you know what this does and why it should be used, please edit this manual.

```
OGLHOME/include
OGLHOME/lib

You have selected to use OpenGL driver.
Specify the correct path (OGLHOME) where OpenGL is installed in your system:
[/usr]
```

Press ENTER unless the OpenGL library is kept somewhere different.

```
G4USE_G3TOG4

If this variable is set, it will be built G3TOG4 module.  Do you want to
set this variable ?  [n]
```

Select ‘yes’ as it will be easier to select it now than have to recompile GEANT4. G3TOG4 stands for GEANT3 to GEANT4.

```
G4ANALYSIS_USE

Activates the configuration setup for allowing plug-ins to analysis tools
based on AIDA (Abstract Interfaces for Data Analysis).  In order to use
AIDA features and compliant analysis tools, the proper environment for these
tools will have to be set (see documentation for the specific analysis tools).
[n]
```

Select ‘yes’ here. AIDA is required for histograms.

```
End of configuration phase.

Creating configuration setup file...

WARNING: the generated configuration file can be edited if necessary!  You
can introduce any change to the configuration file
/home/ward/geant4/geant4.5.2/.config/bin/Linux-g++/config.sh before the
final installation.  To do so, use a shell escape now (e.g.  !vi
/home/ward/geant4/geant4.5.2/.config/bin/Linux-g++/config.sh).

Press [Enter] to start installation or use a shell escape to edit config.sh:
```

Press ENTER.

```
Now starting Geant4 libraries installation...
```

Configuration of GEANT4 is done. The program should now compile. See Appendix A for a listing of the set environment variables. It took about 40 minutes to compile on a 2.5GHz Pentium 4.

5.3 Environment Variables and Directories

After the program has finished compiling, the Configure script needs to be run again without the -install flag.

```
./Configure
```

This will create the files env.csh and env.sh. If using Bash, type

```
source env.sh
```

or for C-shell type

```
source env.csh
```

to set the GEANT4 environment variables as listed in Appendix A. The environment variables must be set before running or compiling GEANT4 applications.

When applications are compiled using the GEANT4 makefiles, the binaries will be placed in $HOME/bin/Linux-g++. This directory must be added to the path.

```
source PATH=$PATH:$HOME/bin/Linux-g++
```

Also, the libraries will be placed in $HOME/geant4/geant4.5.2/lib/Linux-g++ and this directory must be either added to the file /etc/ld.so.conf or LD_LIBRARY_PATH in order for GEANT4 applications to find the shared libraries when they are run.

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$HOME/geant4/geant4.5.2/lib/Linux-g++
```

5.4 Examples

Now that GEANT4 has been installed, it should be tested by running a few examples. The examples can be found in $HOME/geant4/geant4.5.2/examples and one should start with novice/N01. When in the N01 directory, type in

```
make
```

to compile the example and

```
exampleN01
```

to run the example. The tracking information for three geantino-imaginary particles should be displayed on the console.

Now, to test the OpenGL libraries goto the second example, N02. Type

```
make
exampleN02
```

to compile and start the program. Two things should happen. First, an OpenGL window should open showing the geometry of the system. Second, a prompt should appear on the console. Typing

```
/run/beamOn
```

will cause a blue, positive particle to cross the scene. At this point, it is safe to say that the OpenGL portion of GEANT4 is working.

One should now feel free to browse the other examples at leisure. Notice, however, that some of the examples are not well constructed and will crash after a few particles.

6 Installing Java

Java is currently required for the JAIDA and JAS histogram tools. It can be downloaded from reference [J2SE]. The Java development kit software is recommended as well as the runtime engine. The current version of J2SE is 1.4.2.

Since the source is not available, the binary will have to be downloaded and run. Use chmod

```
chmod u+x j2sdk-1.4.2-nb-3.5-bin-linux.bin
```

in order to make the binary executable. Running the binary will start SUN's GUI installation program. This document will assume that Java is installed in $HOME/j2sdk. After installing, the following environment variables must be set

```
export JDK_HOME=$HOME/j2sdk/j2sdk1.4.2
export PATH=$JDK_HOME/bin:$PATH
export LD_LIBRARY_PATH=
$JDK_HOME/jdk/lib/i386:$JDK_HOME/lib/i386/client:$LD_LIBRARY_PATH
```

where the syntax is for the Bash shell. If using C-shell, substitute setenv for export.

7 Installing Histograms

Histograms are the primary drawback of GEANT4. They are difficult to get working and are not as developed as PAW and GEANT3. Of the three choices, only one seemed to work. OpenScientist refused to compile while the source from Anaphe could not be easily obtained and there were problems with the binaries. Only the FreeHEP library was easy to obtain in source and compile.

AIDA is a histogram standard that is designed to replace hbook that was used with PAW. All of the following histogram software are AIDA compliant.

7.1 Open Scientist

The Open Scientist package is downloadable from reference [OpenSci]. It can be downloaded in a compressed format using ftp or by using CVS. The installation directions are not intuitive to Unix users used to './configure && make && make install'. However, the instructions are easy to follow. Unfortunately, the compilation fails due to various compile errors. Because of the difficulty compiling Open Scientist, it was abandoned for use with GEANT4.

Open Scientist was only tested on the RedHat machine using gcc 3.0.2. The version of Open Scientist used was v11r0.

7.2 Anaphe/Lizard

While it is not entirely clear from the documentation, Anaphe is the name of the overall package and Lizard is the name of the analysis application within the package. The home-page of Anaphe can be found at reference [Anaphe].

Anaphe comes with a python script used for downloading the binaries. This is the method the website recommends for downloading and using Anaphe. Unfortunately, this method is very rigid and the correct compiler is required. While an earlier version of the script supported RedHat 7.2, the current version only supported RedHat 7.3.

Anaphe is built with gcc-2.95.2 and gcc-3.2. Neither of these compilers are available on the default RedHat 7.2 system. For Anaphe version 5.0.5, this did not cause it to refuse to load. However, version 5.0.6 complained and refused to work.

Attempting to link a GEANT4 program to Anaphe 5.0.5 causes an instant segmentation fault on the system. Possibly changing to one of Anaphe's recommended compilers might cause this problem to go away. However, GEANT4 should then be re-compiled.

After installing Anaphe 5.0.5, Lizard was able to run on the Debian system. However, on the RedHat system, Lizard did not like LessTIF and required OpenMotif which is difficult to obtain for RedHat 7.2.

The source of Anaphe is hard to get. The CVS information posted on their web-site is incorrect and the source browser is nearly useless. Thus it is recommend that Anaphe not be used until its source code is more available.

7.3 FreeHEP/JAIDA/JAS3

FreeHEP is the name of the overall project [FreeHEP]. JAIDA is an AIDA implementation done in Java that can be connected to a C++ program though the JNI interface. JAS3 is the Java Analysis

Studio [JAS3]. It is build upon JAIDA and is used for analysis of histograms. JAS3 is not currently at production level but is being developed.

The process for installing JAIDA is not simple. Fist, the JAIDA patch for GEANT4 must be obtained and installed according to directions. Next, the FreeHEP libraries must be downloaded and compiled. At this point, GEANT4 is able to produce histograms. However, JAS3 must be installed in order to work with histograms.

7.3.1 Installing JAIDA Patch for GEANT4

The FreeHEP/JAIDA web-site [FreeHEP] claims that 'JAIDA is currently distributed as an overlay product for Geant4. Eventually this will be changed and the normal JAIDA distribution should be usable. Until then, use the special Geant4 Distributions below'. So, in the future patching GEANT4 may not be needed. For the current version of JAIDA, however, this step is important.

First, from the FreeHEP web-site [FreeHEP] select 'JAIDA'. Then find 'GEANT4 and JAIDA'. Download Geant4-JAIDA-3.0.2-Linux-g++.tar.gz and untar it in the GEANT4 directory $HOME/geant4/geant4.5.2. This patch provides a directory called extlib and an updated example placed in $HOME/geant4/geant4.5.2/examples/extended/analysis/A01

```
cd $HOME/geant4/geant4.5.2
tar -zxvf Geant4-JAIDA-3.0.2-Linux-g++.tar.gz
```

Unfortunately, the pre-compiled binaries $HOME/geant4/geant4.5.2/extlib/Linux-g++ are linked to libraries other than on the installation computers. Thus, they need to be recompiled. That is the reason for the next step.

7.3.2 Getting and Compiling Replacement FreeHEP Libraries

CVS is required to get the FreeHEP source. To download the source via CVS follow the instructions from reference [FreeHEP] which are reproduced here.

```
export CVSROOT=:pserver:anoncvs@cvs.freehep.org:/cvs/freehep
cvs login
CVS Password:  jascvs
cvs co -P freehep
```

The syntax above is for the Bash shell. If using the C-shell, substitute setenv for export.

After CVS has finished downloading the sources, type in

```
cd freehep
export FREEHEP=$PWD
```

from the FreeHEP directory. A recommend directory FreeHEP is $HOME/geant4/freehep. However, this recommendation may be safely ignored.

Optionally, to build the Java libraries, type in

```
tools/ant
```

from the FreeHEP directory.

The environment variable $G4INSTALL should be set by the script env.sh in the GEANT4 directory. If not, either re-run env.sh by typing

```
source $HOME/geant4/geant4.5.2/env.sh
```

or set it manually by typing

```
export G4INSTALL=$HOME/geant4/geant4.5.2
```

To compile the C++ libraries, type in

```
export OS=Linux
make
```

from the FreeHEP directory. After compiling, there should be a few new files in the directory $FREEHEP/lib/Linux-g++. These files should be copied to replace the files in $G4INSTALL/extlib/Linux-g++. At this point, GEANT4 should now work with JAIDA.

7.3.3 Running Example A01

Make sure that the environment variables are set as in section 5.3. Change the current directory.

```
cd $HOME/geant4/geant4.5.2/examples/extended/analysis/A01
```

JAIDA has its own environment variables that need to be set. To set the JAIDA environment variables enter

```
source setup-analysis.sh
```

for Bash or

```
source setup-analysis.csh
```

for C-shell. The program can now be compiled using

```
make
```

and run by entering

```
A01app
```

at the command line. The GEANT4 application should start and the ‘JAS AIDA’ window should open. This window should have five different graphs. Entering

```
/run/beamOn
```

should make the plots change as data is entered into them. Now that JAIDA is installed, JAS3 can be installed to display and analyze histograms.

7.3.4 Getting and Running JAS3

JAS3 is obtained from reference [JAS3]. With all the Java environment variables properly set as in section 6, getting JAS3 to run should be as simple as untaring the source and running it:

```
tar -zxvf jas3-Linux-0.7.2.tar.gz
cd jas3-0.7.2
./jas3
```

8 DAWN Postscript

DAWN is used to output images to the postscript format. It is used alongside the OpenGL display.

8.1 Getting and Installing DAWN

DAWN can be obtained from reference [DAWN] and the current version as of writing this document is 3.85d. Download the source file and untar it using the command

```
tar -zxvf dawn_3_85d.taz
```

and change to the new directory. There will be a configure script that is similar to the GEANT4 configure script. Running

```
./configure
```

will cause the script to prompt for input. It asks a few questions and the answers should be

```
C++ compiler :  g++
Optimization level :  -O2
Tcl/Tk window shell :  wish
Xlib include directory :  -I/usr/X11R6/include
Xlib lib directory :  -L/usr/X11R6/lib
OpenGL include directory:  -I/usr/include -I/usr/include/GL
OpenGL lib directory :  -L/usr/lib
OpenGL libraries :  -lGLU -lGL
Socket libraries :
System name :  Default
Remote visualization :  n


Type “make” to start compilation.

After compilation, type “make install” for installation.  Then four executable
files “dawn”, “DAWN_GUI”, “dawnunixd”, and “dawninetd” will be installed
to /home/ward/bin.
```

noticing that the optimization flag uses a letter ‘Oh’ rather than the number ‘zero’. Type

```
make
make install
```

to compile and install DAWN.

8.2 Using DAWN

Before using DAWN, one can set two optional environment variables. A script to set these variables would look like

```
export G4DAWNFILE_DEST_DIR=[directory of GEANT4 application]/dawn/
echo "Set G4DAWNFILE_DEST_DIR=$G4DAWNFILE_DEST_DIR"
export G4DAWNFILE_MAX_FILE_NUMB=1000
echo "Set G4DAWNFILE_MAX_FILE_NUMB=$G4DAWNFILE_MAX_FILE_NUMB"
```

Setting these variables is optional and DAWN will use the present working directory to store its files if these options are not set.

When GEANT4 is running, t the command prompt, type the three commands

```
/vis/open DAWNFILE
/vis/viewer/refresh
/vis/viewer/show
```

to bring up the DAWN GUI. Navigate through the options using the '«' and '»' GUI buttons. Once all the options have been set, press 'OK' to see the image.

9 Using GEANT4 and Environment Variables

Everytime a GEANT4 application is to be run, the environment variables set by env.sh must be set. To set these variables using Bash type

```
source $HOME/geant4/geant4.5.2/env.sh
```

Also, when using the analysis software with JAIDA, the JAIDA script as found with example A01 must be sourced.

```
source setup-analysis.sh
```

10 Conclusion

Now that GEANT4 is installed, refer to the examples and the documentation on the GEANT4 web-site [GEANT4] for more information on how to use the software.

A Environment Variables set by GEANT4's Configure Script

```
On this machine the G4SYSTEM=Linux-g++
On this machine the G4INSTALL=/home/ward/tmp/geant4.5.2
On this machine the G4INCLUDE=/home/ward/tmp/geant4.5.2/include
On this machine the G4TMP=/home/ward/tmp/geant4.5.2/tmp
On this machine the G4LIB=/home/ward/tmp/geant4.5.2/lib
On this machine the G4LEVELGAMMADATA=
/home/ward/tmp/geant4.5.2/data/PhotonEvaporation
On this machine the G4RADIOACTIVEDATA=
/home/ward/tmp/geant4.5.2/data/RadiativeDecay
On this machine the G4LEDATA=/home/ward/tmp/geant4.5.2/../G4EMLOW1.1
On this machine the NeutronHPCrossSections=
/home/ward/tmp/geant4.5.2/../G4NDL3.7
On this machine the CLHEP_BASE_DIR=/home/ward
On this machine the CLHEP_INCLUDE_DIR=/home/ward/include
On this machine the CLHEP_LIB_DIR=/home/ward/lib
On this machine the CLHEP_LIB=CLHEP
On this machine the G4DEBUG=1
On this machine the G4ANALYSIS_USE=1
On this machine the G4UI_BUILD_XAW_SESSION=1
On this machine the G4UI_USE_XAW=1
On this machine the G4UI_BUILD_XM_SESSION=1
On this machine the G4UI_USE_XM=1
On this machine the G4VIS_BUILD_DAWN_DRIVER=1
On this machine the G4VIS_BUILD_OPENGLX_DRIVER=1
On this machine the G4VIS_BUILD_OPENGLXM_DRIVER=1
On this machine the G4VIS_USE_DAWN=1
On this machine the G4VIS_USE_OPENGLX=1
On this machine the G4VIS_USE_OPENGLXM=1
On this machine the OGLHOME=/usr
On this machine the G4LIB_BUILD_G3TOG4=1
On this machine the G4USE_G3TOG4=1
On this machine the G4LIB_BUILD_SHARED=1
On this machine the G4LIB_USE_GRANULAR=1
```

B Environment Variables set by JAIDA's Configure Script

```
JDK_HOME set to /home/ward/j2sdk/j2sdk1.4.2
JVM_ARGS set to -Dhep.aida.IAnalysisFactory=jas.aida.gui.  JASGUIAnalysisFactory
CLASSPATH set to /home/ward/geant4/geant4.5.2/extlib/java/JASAIDA.jar:
/home/ward/geant4/geant4.5.2/extlib/java/aida.jar:
/home/ward/geant4/geant4.5.2/extlib/java/aida-dev.jar:
/home/ward/geant4/geant4.5.2/extlib/java/freehep-hep.jar:
/home/ward/geant4/geant4.5.2/extlib/java/freehep-base.jar:
/home/ward/geant4/geant4.5.2/extlib/java/openide-lookup.jar:
/home/ward/geant4/geant4.5.2/extlib/java/jas.jar:
/home/ward/geant4/geant4.5.2/extlib/java/jel.jar
LD_LIBRARY_PATH set to /home/ward/geant4/geant4.5.2/extlib/Linux-g++:
/home/ward/j2sdk/j2sdk1.4.2/jre/lib/i386/client:
/home/ward/j2sdk/j2sdk1.4.2/jre/lib/i386:
/home/ward/geant4/geant4.5.2/extlib/Linux-g++:
/home/ward/j2sdk/j2sdk1.4.2/jre/lib/i386/client:
/home/ward/j2sdk/j2sdk1.4.2/jre/lib/i386:
/home/ward/j2sdk/j2sdk1.4.2/jdk/lib/i386:
/home/ward/j2sdk/j2sdk1.4.2/lib/i386/client:
```

```
/home/ward/geant4/geant4.5.2/lib/Linux-g++:
/home/ward/lib:
/home/ward/Anaphe/installation/specific/redhat72/gcc-2.95.2/5.0.3/lib
G4ANALYSIS_AIDA_CONFIG_LIBS set to -L/home/ward/geant4/geant4.5.2/extlib/Linux-g++
-lAIDAJNI -lFHJNI -L/home/ward/j2sdk/j2sdk1.4.2/jre/lib/i386/client -ljvm
```

C Example .bash_profile File

```
# .bash_profile
# Get the aliases and functions
if [ -f  /.bashrc ]; then
.   /.bashrc
fi

PATH=$HOME/bin:$PATH:$HOME/bin/Linux-g++
LD_LIBRARY_PATH=$HOME/geant4/geant4.5.2/lib/Linux-g++

#Java Stuff
JDK_HOME=$HOME/j2sdk/j2sdk1.4.2
PATH=$JDK_HOME/bin:$PATH
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JDK_HOME/jdk/lib/i386
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$JDK_HOME/jdk/lib/i386/client

export PATH
export LD_LIBRARY_PATH
export JDK_HOME
```

References

[OpenSci] Barrand, G. *Open Scientist Web-site* http://www.lal.in2p3.fr/OpenScientist/

[DAWN] Tanaka, S. *DAWN Web-site* http://geant4.kek.jp/∼tanaka/

[Anaphe] *Anaphe Web-site* http://anaphe.web.cern.ch/anaphe/

[CLHEP] *CLHEP Web-site.* http://wwwasd.web.cern.ch/wwwasd/lhc++/clhep/

[FreeHEP] *FreeHEP Web-site.* http://java.freehep.org/index.html

[GEANT4] *GEANT4 Web-site.* http://geant4.web.cern.ch/geant4/

[JAS3] *JAS3 Web-site.* http://jas.freehep.org/jas3/index.html

[J2SE] *Java 2 Platform, Standard Edition (J2SE) Home-page.* http://java.sun.com/j2se/

All web-sites are accurate as of 25 August 2003.