

ILC Detector Simulation FAQ

The ILC Detector Simulation FAQ

General

Why are detector simulations needed?

ILC Detector simulations are needed to

- optimize full detector designs for physics performance on benchmark processes
- optimize the designs of subsystems and subdetectors
- compare proposed detector technologies with each other (in concert with test beam)
 - calorimeter examples - RPC, GEM, scintillator, Silicon, lead tungstate, hybrid
- compare different full detector designs
- compare Monte Carlo results with test beam for Geant4 physics list validation
- compare, test, and debug reconstruction and analysis algorithms and other software
- understand the role of dead materials in affecting the overall detector performance
- model the effects of hardware inefficiencies such as cross-talk, dead cells, etc.
- understand the effects of backgrounds on detector performance
- establish a software infrastructure for the actual detector experiment(s)

What groups are involved with ILC detector simulations?

- [ACFA](#)
- [ALCPG](#)
- [CALICE](#)
- [ECFA](#)
- [Linear Collider R&D Working Group \(LCRD\)](#)
- [NICADD](#)
- [University Consortium for Linear Collider R&D \(UCLC\)](#)

Which labs are the primary ones involved with ILC detector simulations?

- [SLAC](#)
- [DESY](#)
- [KEK](#)
- [FNAL](#)
- [IN2P3](#)

What (US) schools and universities are involved with ILC detector simulations?

- [University of Colorado at Boulder](#)
- [University of Kansas](#)
- [University of Oregon](#)
- [Cambridge](#)
- [Iowa State](#)
- [Northern Illinois University](#)

Where can I get additional information about software packages?

The [LCSim Software Index](#) has links to all the primary tools for ILC Detector Simulations. The [Portals Section](#) lists sites that cover full suites of software.

What conference cover ILC detector simulations?

[LCWS](#)
[ALCPG Workshops](#)
[ECFA workshops](#)
[ACFA Workshops](#)

Where can I find documents about ILC detector simulations aside from workshop talks?

[ILC Wiki](#)
[NICADD agenda server](#)
[ILC Agenda](#)
[LCNotes](#)

Is there a more complete list of ILC groups/links?

[List of ILC Groups, Conferences, and other Resources](#)

Feedback

Is there an forum for ILC detector simulation issues?

The [linearcollider.org forum](#) is an appropriate place to post your questions on detector simulations and reconstruction and analysis.

Is there a regular meeting that covers ILC detector simulation software?

The [weekly LCD software meeting](#) for discussing all aspects of detector simulations and analysis and reconstruction. This meeting takes place every Tuesday at 1:30 PST, unless cancelled.

Are there mailing lists for questions or announcements about ILC detector simulation?

[lcd-l](#)
[lcd-dev](#)
[calice-sw](#)

Detector Concepts

What are the proposed detector concepts?

There are now [four proposed detector designs](#) for the ILC, each of which has an an outline document.

(above based on slide from Hitoshi Yamamoto)

What is the SiD?

- [Silicon Detector \(SiD\)](#) (American origin)
 - Silicon tracker, 5T field
 - SiW ECAL
 - 4 'coordinators' (2 Americans, 1 Asian, 1 European)

From the SiD homepage, "The Silicon Detector Design Study is being organized to develop the SiD Detector Concept into a detailed, optimized, and fully integrated detector design on a roughly one year time scale. The SiD concept incorporates Si/W calorimetry and Si tracking in a Linear Collider detector design which attempts to optimize physics performance while realistically constraining costs."

[SiD Detector Outline Document](#)
[SiD Meeting Agendas](#) - with documents

What is the GLD?

[GLD Detector Outline Document](#)

[GLD Detector Outline Document Kickoff](#)

- [GLD](#) (Asian origin)
 - TPC (+Large Silicon IT), 3T field
 - W/Scintillator ECAL ("large" radius)
 - 6 'contact persons': (2 Americans, 2 Asians, 2 Europeans)

What is the LDC?

- [Large Detector Concept \(LDC\)](#) (European origin)
 - TPC (+Silicon IT), 4T field
 - SiW ECAL ("medium" radius)
 - 6 'contact persons': (2 Americans, 2 Asians, 2 Europeans)

[LDC Detector Outline Document](#)

What is the fourth concept?

- [fourth concept](#)

[Fourth Concept Outline Document](#)

Subsystems

Where can I go for information about ILC detector R&D programs?

[LCRD UCLC FY06](#)
[ILC Detector R&D](#) - mostly archival materials

What are the main subsystems of an ILC detector?

vertexing
forward tracking
central tracking
calorimetry (ECAL, HCAL)
muon system
magnet
DAQ, electronics, detector integration

(from slide of Ties Behnke)

Calorimetry

What technologies are being considered for the HCAL?

RPC - Stainless Steel (SS)
GEM - SS
scintillator - SS
DREAM
hybrid

What technologies are being considered for the ECAL?

Si - W
Si - Pb
scintillator - W, Pb
lead tungstate crystals
hybrid

What technologies are being considered for the MUON subsystem or tailcatcher?

RPC
scintillator

Tracking

What technologies are being considered for the tracking system?

Silicon
TPC

Vertexing

What technologies are being considered for the vertexing system?

Forward Region

luminosity monitor
beam monitor
photon calorimeter

Software Frameworks

What tools are available for detector simulation and related research?

Program	Description	Detector	Language	IOFormat	Region
Simdet	fast Monte Carlo	TeslaTDR	Fortran	StdHep/LCIO	EU
SGV	fast Monte Carlo	simple Geometry, flexible	Fortran	None (LCIO)	EU
Lelaps	fast Monte Carlo	SiD, flexible	C++	SIO, LCIO	US
Mokka	full simulation - Geant4	TeslaTDR, LDC, flexible	C++	ASCI, LCIO	EU
BrahmsSi	Geant3 - full simulation	TeslaTDR	Fortran	LCIO	EU
SLIC	full simulation - Geant4	SiD, flexible	C++	LCIO	US
LCDG4	full simulation - Geant4	SiD, flexible	C++	SIO, LCIO	US

Jupiter	full simulation - Geant4	JLD (GDL)	C++	Root (LCIO)	AS
BrahmsRec	reconstruction	TeslaTDR	Fortran	LCIO	EU
Marlin	reconstruction	Flexible	C++	LCIO	EU
hep.lcd	reconstruction	SiD (flexible)	Java	SIO	US
org.lcsim	reconstruction	SiD (flexible)	Java	LCIO	US
JupiterSatellites	reconstruction	JLD (GDL)	C++	Root	AS
LCCD	Conditions Data Toolkit	All	C++	MySQL, LCIO	EU
GEAR	Geometry description	Flexible	C++ (Java?)	XML	EU
LCIO	Persistency and data model	All	C++, Java, Python, FORTRAN	SIO, LCIO	AS,EU,US
JAS3/WIRED	Analysis Tool / Event Display	All	Java	LCIO, StdHep, HepRep	US,EU

(from slide of Tony Johnson)

Isn't there one main software framework for ILC detector simulation that I can use?

Each of the major detector concepts uses a separate software framework, with LCIO providing interoperability.

What frameworks are available for ILC Detector Simulations?

There are four different toolchains that can be used, corresponding to the LDC, SiD, GLD, and "fourth detector" concepts. The software suites corresponding to the first two detectors can simulate all of the detector concepts, so there is no strict division on which tools should be used for a certain detector study.

What are the software frameworks that correspond to each design?

Detector	Software
GLD	Jupiter, JupiterSatellites, Root
LDC	Mokka, Marlin, MarlinReco, CED
SiD	SLIC, org.lcsim, JAS3, WIRED
4th	AliRoot

Which software tools are associated with SiD?

The [ALCPG](#) and [SLAC](#) use the [SLIC Simulations Package](#) (C++) with the [org.lcsim](#) for reconstruction and analysis using Java. This system also uses the [JAS3](#) graphical analysis environment. Much of the functionality within org.lcsim comes from the [FreeHep Java Library](#).

Which software tools are associated with GLD?

Finally, the [ACFA](#) group has a [suite of software](#) based on the [ROOT](#) framework.

Which software tools are associated with LDC?

There is a C++ framework developed under [ECFA](#) that uses [Mokka](#) for the simulator with the [MARLIN](#) tool for analysis and reconstruction.

[ILC Software Meeting at Cambridge](#) - focuses on LDC framework (Mokka, Marlin)

What software is associated with the fourth concept?

The 4th concept is using ILCROOT, which is itself based on AliRoot.



The remainder of this FAQ focuses on the US/ALCPG/SLAC software suite based on SLIC and org.lcsim. There is also some basic discussion of Mokka and Marlin and CALICE software. I do not discuss the GLD or 4th concept frameworks. --JM

Bug Reports

How do I file a bug report or make a feature request related to ILC software?

The [SLAC Jira Bugtracker](#) allows users to submit bug reports or make feature requests.

The projects relevant to ILC.

- [SLIC](#)
- [LCDD](#)
- [GeomConverter](#)
- [LCDetectors](#)
- [LCIO](#)
- [LCSIM](#)
- [www-lcsim](#)

Most of these packages will be covered in more detail within this FAQ.

How do I file a bug report or make feature request for auxiliary ILC software?

Several packages are used by org.lcsim that are not specific to ILC simulations. There is a [separate FreeHep bugtracker](#) for these projects.

- [JAS3](#)
- [AIDA](#)
- [WIRED](#)

How do I get an account on the ILC or FreeHep Jira?

Contact Tony Johnson <tony_johnson@slac.stanford.edu> to obtain an account on either of these systems.

Software Repositories

How can I checkout projects anonymously from the SLAC CVS?

You can checkout projects with this command.

```
cvs -d :pserver:anonymous@cvs.freehep.org:/cvs/lcd co [module]
```

The *module* is the name of the package to checkout of CVS.

What modules are available from the SLAC CVS?

These are the main modules.

Module	Description
GeomConverter	Java geometry package for org.lcsim
ExampleMavenProject	Simple project showing how to use Maven
LCDetectors	Repository of Compact Detector Descriptions
LCPhys	The Linear Collider Physics List for Geant4
SimDist	Standalone build of simulation framework (Geant4/CLHEP,Xerces,LCIO,GDML/LCDD,LCPhys,SLIC)
SlacMCPrj	Backups of Unix software dist (mostly scripts)
SlicDiagnostics	An org.lcsim-based suite of diagnostics histograms for LCIO files
lcio	The Linear Collider IO project
lcdd	The LCDD geometry project
www-lcsim	The lcsim.org website
slic	The SLIC simulation package

How do I get an account at the SLAC CVS?

Contact Tony Johnson <tony_johnson@slac.stanford.edu> to request a SLAC CVS account.

Where can I go for more information about the SLAC CVS?

[SLAC CVS page](#)

Can I browse the SLAC CVS on the WWW?

[jcvs](#)

What are some other source code repositories I should know about?

[Freehep CVS](#), [Freehep Maven Repositories](#)
[LCIO](#)
[CALICE](#)
Marlin, MarlinReco, CED
GDML

What are the recommended CVS tools for Windows users?

Windows users should look at [Tortoise CVS](#) which adds extensions to Explorer for easily checking out and managing CVS projects.

Geant4 and CLHEP

What is Geant4?

[Geant4](#) is the core Monte Carlo simulator used by all ILC detector simulations programs. It tracks the passage of particles through matter and simulates the particle interactions.

How do I build Geant4?

Before installing Geant4, you need to download and build <http://proj-clhep.web.cern.ch/proj-clhep/>.

Then download the Geant4 sources from the [distribution page](#).

Once you have obtained the sources, please read the [Installation Guide](#) carefully.

The following sequence of commands should get you up and running on Linux (using bash shell).

```
export CLHEP_BASE_DIR=/your/path/to/clhep
tar -zxvf geant4.x.tar.gz
cd geant4.x
export G4INSTALL=$(pwd)
export G4WORKDIR=$(pwd)
export G4SYSTEM=Linux-g++
cd geant4.x/source
make
make includes
cd ../physics_lists/hadronic
make
```

This will take up to several hours depending on the speed of your machine.

Next, try to build an example application.

```
cd geant4.x/examples/novice/N03
export G4WORKDIR=$(pwd)
make all
```

Now you should have the `./bin/Linux-g++/N03` binary.

How do I rebuild just the Geant4 visualization libraries?

After reconfiguring your environment using the G4VIS variables, rebuild the Geant4 visualization as follows.

```
cd $G4INSTALL/source/visualization
gmake clean
gmake
cd $G4INSTALL/source/interfaces
gmake clean
gmake
cd $G4INSTALL/source
gmake libmap
```

SLIC

What is SLIC?

SLIC stands for "Simulator for the Linear Collider". It is a full simulation package that uses the [Geant4](#) Monte Carlo toolkit to simulate the passage of particles through the detector. SLIC outputs [LCIO](#) files that can be analysed using a variety of frameworks and language bindings. SLIC uses a separate backend for the input of detector data called Linear Collider Detector Description ([LCDD](#)), which itself is based on the [GDML](#) project from CERN.

Where is SLIC's homepage?

The [SLIC Confluence page](#) is the best source for up-to-date information on this software package.

Who is the primary maintainer of SLIC?

Jeremy McCormick <jeremym@slac.stanford.edu> is the primary author and maintainer of SLIC. The handling of MCParticles in SLIC was based on earlier work by Ron Cassell <cassell@slac.stanford.edu> from a package called Linear Collider Simulator (LCS). Much work has also been done by Ron to verify and debug the LCIO output files.

Where should I go for news or announcements about SLIC?

At which HEP sites and labs is SLIC installed?

Fermilab: Here are [instructions for Fermilab users](#) on getting started with SLIC.

SLAC: Here are [instructions for SLAC users](#) on running slic from the [public Unix machines](#) at the lab. SLIC can be used on any of the load-balanced interactive Linux clusters, including noric, iris, and yakut.

NICADD: A copy of SLIC is available on the [NICADD](#) Linux clusters at `/k2data/APPS/SimDist`.

Installing SLIC

Which operating systems and compilers does SLIC support?

These are known working configurations of platform and compiler.

platform	compiler	compiler version
Linux	g++	3.3, 3.4
OSX	g++	4.0
Windows (Cygwin)	g++	3.4

The only supported compiler is **g++**. Other versions may work, but these are the ones that are known to build successfully.

How do I get simulation binaries for my platform?

SimDist downloads contain binaries for Linux, OSX, and Windows.

These are located at ...

```
http://www.lcsim.org/dist/slic/
```

To download a Linux release, use these commands.

```
wget http://www.lcsim.org/dist/slic/slic-1_13_3-Linux-g%2B%2B-bin.tar.gz
tar -zxvf slic-1_13_3-Linux-g++-bin.tar.gz
```

What is SimDist?

The *SimDist* package bundles together all the dependencies of SLIC into one project. This greatly simplifies the build process.

How can I compile simulation binaries for my platform?

Follow the [build instructions for the SimDist project](#) to make binaries for your platform.

How do I build SLIC and the packages it depends on from scratch?

Building SLIC and its dependencies "from scratch" is not encouraged now that the SimDist project can manage this build procedure.

There are separate instructions for [building on Linux](#) and [building on Windows](#) if you must do this.

Running SLIC

How do I run if I installed with SimDist?

SimDist has a run script at *scripts/slic.sh* which will setup the LD_LIBRARY_PATH to contain the Xerces lib directory and execute the current slic binary from the *packages/slic/\$SLIC_VERSION* directory.

It is useable from anywhere on the host where it was installed.

```
/path/to/my/SimDist/scripts/slic.sh [options]
```

This script passes all arguments to *slic*, so use it as you would the normal binary.

How do I run the simulator binary from the slic home directory?

Assuming you are within the slic package's top directory you would run it with this command.

```
./bin/$G4SYSTEM/slic [options]
```

The G4SYSTEM variable depends on your platform. For instance, on Linux, it is set to *Linux-g++*, so the command would be ...

```
./bin/Linux-g++/slic [options]
```

Does SLIC have any runtime dependencies?

If you installed SLIC in the recommended way, the only runtime dependency would be the [Xerces C++](#) shared library. The Xerces library directory needs to be added to the LD_LIBRARY_PATH (bash), so that the loader can find this library when slic executes.

```
export LD_LIBRARY_PATH=/path/to/my/xerces/install/lib:$LD_LIBRARY_PATH
```

If you see this error, it means that the LD_LIBRARY_PATH has not been set correctly.

```
./slic/bin/Linux-g++/slic: error while loading shared libraries: libxerces-c.so.27: cannot open shared object file: No such file or directory
```

The SimDist run script will set this variable automatically.

How do I get help on the command-line syntax?

To see explanations of slic's command-line options, use the "-h" option.

```
slic -h
```

This will print slic usage and exit.

What command-line options does SLIC support?

Typing *slic -h* will print the available command-line options.

```
*****
* Command Line Options *
*****
```

Option	Full Name	Min Args	Max Args	Macro Command	Description
-h	--help	0	0	/slic/usage	Print SLIC usage.
-n	--interactive	0	0	/control/interactive	Start a Geant4 interactive session.
-v	--version	0	0	/slic/version	Print SLIC version.
-m	--macro	1	1	/control/execute	Execute Geant4 commands from a file.
-g	--lcdd-url	1	1	/lcdd/url	Set LCDD geometry file URL.
-i	--event-file	1	1	/generator/filename	Set event input file full path.
-o	--lcio-file	1	1	/lcio/filename	Set name of LCIO output file.
-p	--lcio-path	1	1	/lcio/path	Set directory for LCIO output.
-O	--autoname	0	1	/lcio/autoname	Automatically name the LCIO output file.
-x	--lcio-delete	0	0	/lcio/fileExists delete	Delete an existing LCIO file.
-r	--run-events	1	1	/run/beamOn	Run # of events.
-s	--skip-events	1	1	/generator/skipEvents	Set number of events to skip.
-l	--physics-list	1	1	/physics/select	Set Geant4 physics list.
-L	--log-file	0	0	/log/filename	Set logfile name.
-d	--seed	0	1	/random/seed	Set the random seed. (No argument seeds with time.)

Each option is mapped in a straightforward way to a corresponding Geant4 UI command, so the same exact commands can be executed either from the command prompt or in the Geant4 interactive terminal.

How do I find out about the SLIC version I am running?

The command

```
slic -v
```

prints the SLIC version, along with a lot of other information (probably too much).

```
Simulator for the Linear Collider; SLIC; v1r13p6; Jeremy McCormick and Ron Cassell; SLAC; Tue Apr 4 17:32:07 PDT 2006
```

The `/slic/version` command will do the same from a macro or UI terminal.

How do I get detailed information about the SLIC version I am running?

SLIC prints a splash screen as it starts up with the version, build date, and other information.

```
*****
App      : Simulator for the Linear Collider (SLIC)
Version  : v1r13p6
Date     : Tue Apr 4 17:32:07 PDT 2006
Authors  : Jeremy McCormick and Ron Cassell
Inst     : SLAC
WWW      : http://www.lcsim.org/software/slic
Contact  : jeremym@slac.stanford.edu
*****
```

How do I find out the version of Geant4 that SLIC was built with?

The Geant4 toolkit prints a splash screen that SLIC displays when it starts up.

```
*****
Geant4 version Name: geant4-08-00-patch-01    (10-February-2006)
                    Copyright : Geant4 Collaboration
                    Reference  : NIM A 506 (2003), 250-303
                    WWW       : http://cern.ch/geant4
*****
```

The Geant4 version in the figure above is 8.0.p01.

How do I get help on SLIC's custom Geant4 commands?

SLIC commands are added into various directories within the Geant4 UI hierarchy. From interactive mode, use this command to print information on a SLIC or LCDD command.

```
help [command]
```

Those directories containing only SLIC or LCDD commands are marked with *SLIC* or *LCDD*.

```
/lcio/    LCIO output commands. [SLIC]
```

Type

```
help [dir]
```

or

```
help [command]
```

for information about the user interface directory or command. Of course, this requires that the programmer actually put in some useful help string. Most commands and directories have such information.

How do I run a macro containing UI commands?

You can put Geant4 UI commands into one or more macro files that slic can execute. This is done with the "-m" switch. There can be any number of these switches given to slic at the commandline. For instance, this command will execute the two macros *init.mac* and *run.mac*.

```
slic -m init.mac -m run.mac
```

The macros are executed in the order given at the command-line.

If macros are interspersed throughout the command switches, then SLIC will also execute each command in order.

```
slic -m init.mac -x -o output -p myDir -m run.mac
```

First *init.mac* will be executed. Then the three commands corresponding to the "-xop" switches and finally *run.mac* presumably executes the */run/beamOn* command.

How do I run in interactive mode?

SLIC can be started in interactive mode using the *-n* option or the */control/interactive* UI command.

This command will start Geant4 in *PreInit* mode where the geometry is not loaded.

```
slic -n
```

Other commands may also be executed before the interactive mode starts.

```
slic -g myGeom.lcdd -n
```

The LCDD file will be constructed, so the simulator will start in *Idle* mode.

What is a typical command to run SLIC in batch mode?

The following command illustrates some of the options that could be used in a typical batch run.

```
slic -g myGeom.lcdd \           # geometry file
-i events.stdhep \             # StdHep input file
-p lcio/ \                     # path for LCIO output file
-o output.slcio \              # name of LCIO output file
-x \                           # delete existing LCIO file
-r 1234 \                      # seed the random engine
-s 100 \                       # number of events to skip
-r 1000                        # number of events to run
```

How can I check on the progress of my SLIC job(s)?

In bash, direct all output to a file, as follows.

```
slic &> job.log &
```

Now use the tail command to look at the output file as the job progresses.

```
tail -f job.log
```

The *BeginEvent* and *EndEvent* markers are the best indicators of the job's progress.

Most batch systems will provide logging services, and the *tail* command can be used on these files, too.

How do I run SLIC on the LSF batch system?

The *bsub* command is used to run SLIC on LSF.

This is an example of submitting a job to run 1000 events through the sid00 detector on the xlong batch queue.

```
bsub -q xlong -o `pwd`/job.log -e `pwd`/err.log time slic -g sid00.lcdd -i events.stdhep -o events_sid00 -r 1000
```

LSF accepts plain bash scripts to start jobs. Mine usually use a few for loops to run combinations of geometries with StdHep files.

```
for g in $(ls *.lcdd); do
  for s in $(ls stdhep/*.stdhep); do
    fname=$(basename ${s%.*})_${basename ${g%.*}}
    bsub -o `pwd`/$fname.log -e `pwd`/$fname.err -q xlong time slic -g $g -i $s -o $fname
  done
done
```

The *stdhep* directory should contain a symlink to the StdHep files used for input events.

Example LSF submission scripts can be found in Jeremy McCormick's jobs directory on the SLAC Linux machines at *~jeremym/work/jobs*.

How can I make SLIC look for XML schemas in a local directory?

The current directory is always searched for the XSD files, first.

The GDML_SCHEMA_DIR environment variable may point to a single directory where all XSD files should be located. LCDD uses GDML for entity resolution, also, so the LCDD schemas can be placed in this directory, also.

How do I install the XML schemas to a local directory?

From the lcdd directory, execute *make schema_install* to copy all the GDML and LCDD schemas in the *schemas* directory to another existing directory.

How can I automatically set the name of the LCIO file?

The *-O* switch turns on SLIC's auto-naming feature, which will give the output file a reasonable name based on the event generator used and the name (or tag) of the geometry. The */lcio/autoname* command is the corresponding UI command.

How can I customize SLIC's autonaming feature?

The autoname feature can be customized by providing arguments to the `/lcio/autoname` command. These specify fields that should be inserted in the automatically generated filename.

Here is a summary of these options.

field	description
binary	basename of the slic binary
application	the string 'SLIC'
version	SLIC's version string from VERSION
geometry	geometry tag from the LCDD header
date	current date
event	description of the current event from the generator
eventNumber	number of events
run	run number

The following command illustrates the autoname feature.

```
/lcio/autoname event application version geometry
```

It would result in an automatically-generated name similar to *neutrons_SLIC_v1r13p6_myGeometry*.

How can I automatically delete an existing LCIO file?

The `-x` option or `/lcio/fileExists delete` option will cause an existing LCIO file to be overwritten.

How do I skip events in a StdHep file?

The `-s` option or `/generator/skipEvents` command take the number of events to skip as an argument. This is only applicable to file-based event sources such as StdHep or LCIO files.

How do I set the random seed?

The `-r` option or the `/random/seed` command controls the CLHEP random seed setting. Without an argument, the current time is used as a seed. Otherwise, an integer argument is used as the seed.

How do I dump the state of the CLHEP random engine?

Dumping random state is a built-in feature of Geant4.

Turn on the save feature.

```
/random/setSavingFlag true
```

This will cause the random state for each event to be saved to *currentEvent.rndm*.

To save the current event to a unique file, use this command.

```
/random/saveThisEvent
```

This copies *currentEvent.rndm* to *runXXXevtYYY.rndm*.

To do this for every event, put the *saveThisEvent* command in a macro called *rndmSave.mac* and run events in this fashion.

```
/run/beamOn 1000 rndmSave.mac
```

This will cause the random state to be dumped for every event.

How can I set the directory for the random dumps?

The `/random/setDirectoryName` command sets the base directory where the rndm files will be placed.

How do I load back the state of the random engine that I dumped previously?

To restore the random state, use this Geant4 UI command.

```
/random/resetEngineFrom currentEvent.rndm
```

This can be used to "replay" events of interest. This feature will probably be used with SLIC's skipEvents feature in order to jump to the event first before restoring its random state.

```
/random/resetEngineFrom run0event1000.rndm  
/generator/skipEvents 1000
```

How can I see which physics lists are available?

Execute the command `/physics/printLists` to see the available Geant4 physics lists.

How do I set the physics list?

The `-l` option or `/physics/select` command takes the name of the physics list to use as an argument.

```
/physics/select LHEP
```

This must happen in the *PreInit* stage. Once the simulator is in *Idle* state, the physics list is set permanently for that run, and SLIC must be restarted to use a different list.

How can I customize Geant4's stepping in the magnetic field?

A number of options are available for tweaking the handling of transportation through the magnetic field in the `/field` command directory.

The command `selectStepper` sets the integrator, where the argument can be one of the following.

```
G4ClassicalRK4  
G4ExplicitEuler  
G4HelixExplicitEuler  
G4HelixHeum  
G4HelixImplicitEuler  
G4HelixSimpleRunge  
G4ImplicitEuler  
G4CashKarpRK45  
G4SimpleHeum  
G4SimpleRunge
```

The command `setDeltaOneStep` sets the maximum stepsize that will be taken in the field.

The command `setDeltaIntersection` sets the maximum distance from the chord line.

Refer to the [Electromagnetic Fields Chapter of the Geant4 Application Guide](#) for more details on these settings.

How do I set the maximum number of events?

The `-r` switch sets the number of events to run, and it corresponds to Geant4's built-in `/run/beamOn` command.

What event generators are available from within SLIC?

SLIC can read [StdHep](#) files using Willy Langeveld's lightweight XDR interface from [Lelaps](#).

An MCParticles collection of an LCIO file can also be used as a source of events, which is useful for repeating events run in other simulation frameworks.

How do I use a StdHep file?

To select a StdHep file with physics events, use the `-i` switch or `/generator/filename` command followed by an absolute or relative path to the file. Any file with an extension of `.stdhep` or `.xdr` will be automatically interpreted as a StdHep file.

How do I use an LCIO file as an event source?

SLIC also accepts LCIO files as event sources. Use the commands as if the file were a StdHep source, and the LCIO generator will be automatically selected if the file has the extension of .slcio.



The LCIO event generator feature is an experimental feature that has not undergone much testing (e.g. There be bugs here!).

How do I use the General Particle Source (GPS)?

Geant4 includes a flexible and powerful particle generator called the General Particle Source or GPS.

Execute this command to select GPS as the event source.

```
/generator/select gps
```

Actually, GPS is the default, but this can be used to switch if another generator is being used (e.g. in interactive mode).

Refer to the [GPS User's Guide](#) for detailed instructions on using this tool.

There are also a few sample GPS macros in the slic CVS project at *macros/gps*.mac_*.

How do I use the particle gun?

Geant4 includes a simple particle gun that can be selected with this command.

```
/generator/select gun
```

However, GPS is the recommended single particle source, and the G4ParticleGun should be considered deprecated, in my opinion.

How can I print information about the current event generator?

Use the command */generator/dumpEvent* to display information about the current generator and its event.

What GUIs are available for SLIC?

There are only a limited number of

[SLIC Runner](#) is a Java program that uses Swing.

TODO: Fix SLIC Runner to work with latest SLIC.

The built-in GUIs of Geant4 are available, including the XM interface.

If I have a problem with SLIC or find a bug, what information should I include in my report?

Please include the following information in bug reports.

- Your name and institutional affiliation
- Operating system and version
- Version of gcc compiler

```
gcc -v
```

- Versions of all SimDist software packages

```
cat SimDist/packages/*/VERSION &> myversions.txt
```

- If not using SimDist, please provide at least the SLIC and GEANT4 version numbers.
- Run macro for reproducing the bugged LCIO file
- Publically accessible location of the StdHep input file (if any)
- Versions of GeomConverter and org.lcsim that you are using (if applicable)

Geometry using GDML and LCDD

Why would you use XML for describing detector geometry?

- naturally hierarchical

- platform independent
- human editable
- extendable
- described by a standardized and powerful schema language (XML Schema or XSD)
- generated or transformed easily from nearly any programming language (C++, Java, Python, etc.)
- enjoys broad support and acceptance
- overall a good fit for describing detector data

What is GDML?

GDML stands for Geometry Description Markup language. It is an XML format for the description of CSG geometries with a primary binding to the Geant4 toolkit. There is also a [Python](#) binding with [Root](#). GDML allows geometry data to be interchanged between different programs, such as simulators, analysis and reconstruction software, and visualization tools.

What features does GDML support?

GDML defines a document structure and a list of legal elements.

- defines
 - constants, reflections, positions, rotations
- materials
 - material, isotope, element, mixture
- solids
 - box, sphere, tube, cone, polycone, parallelepiped, trapezoid, torus, polyhedra, hyperbolic tube, elliptical tube, ellipsoid
 - boolean solids
- volumes
 - assembly volumes and reflections
 - replicas and divisions
 - parameterised volumes (position, rotation and size)

Where can I go for more information on GDML?

The current, primary maintainer of GDML, Witold Pokorski, made a [good overview presentation of GDML](#) for the [CHEP 2006](#) conference.

Browse the [GDML CVS online](#) or use the [LXR browser](#).

A supplementary page is maintained by Jeremy McCormick at <http://www.lcsim.org/software/gdml>.

What is LCDD?

LCDD is the Linear Collider Detector Description markup language, and is itself based on the GDML toolkit.

Why should I use LCDD?

The LCDD geometry backend is designed for prototyping, debugging, and experimenting with different detector geometries, material definitions, readouts, and physics settings. There are many options to explore in the Geant4 toolkit, so having this type of "Swiss Army Knife" tool saves a lot of (programming) time.

LCDD provides a 100% runtime description of the detector and its associated properties, including the detailed volume hierarchy, assignment and properties of sensitive detectors, and the specification of detector component identifiers, among many other features. In practice, writing files in this standardized format is much easier than trying to author custom C++ code for each proposed full detector design and their subdetectors.

What features does LCDD support?

- volumes
 - logical volumes
 - assignment of sensitive detector, visualization attributes, region, physics lists
 - physical volumes
 - volumes identifiers
- sensitive detectors
 - tracker
 - calorimeter
 - readout binning and segmentation
- magnetic field
 - simple solenoid
 - RZB map
- visualization attributes
- region
- physics limits
- identifiers
 - 32 and 64 bit ids
 - identifier dictionary

Where can I go for more information on LCDD?

[LCDD Homepage](#)

Where can I find examples of LCDD?

The directory *examples* in the slic CVS project contains a number of hand-coded LCDD example files.

What is the best way to create LCDD files?

Use the compact description.

Or hand-code them.

How do I dump the current Geant4 geometry to GDML using SLIC?

```
/lcdd/dumpGDML mygeom.gdm1
```

How do I read an LCDD file using SLIC?

LCDD can read files either from the local file system or from a URL. The file location can be set with the *-g* option and its corresponding command */lcdd/url*.

This is an example of reading a local file from the command line.

```
slic -g /path/to/myGeom.lcdd
```

This command reads the geometry from a URL.

```
slic -g http://www.lcsim.org/detectors/sid00/sid00.lcdd
```

If the *-g* option is specified from the command-line, SLIC assumes that the simulator should be initialized to the *Idle* state after the geometry is loaded.

How do I read a GDML file using SLIC?

The LCDD system supports reading both plain GDML and LCDD files from SLIC.

To read in a "plain" GDML file, simply use the *"-g"* option as you normally would ...

```
slic -g myGeom.gdm1
```

The LCDD system will use the GDML schema instead of LCDD and the geometry should be loaded successfully.

How do I check my Geant4 geometry for overlaps?

```
/geometry/test/recursive_test
```

Detector and Event Visualization

How can I use Geant4 to visualize the detector?

The Geant4 OpenGL visualization driver can be used to immediately visualize the detector and events. The following command can be used to visualize the detector geometry plus hits and trajectories from the event.

```
/vis/scene/create  
/vis/scene/add/volume  
/vis/scene/add/hits  
/vis/scene/add/trajectories  
/vis/open OGLSX
```

There is a similar macro in the slic directory at *macros/vis_gl.mac*.

How can I make a HepRep file using Geant4?

A command similar to this will write out a HepRep zip file to the current directory.

```
/vis/scene/create
/vis/open HepRepXML
/vis/viewer/set/culling global false
/vis/scene/add/volume
/vis/scene/add/trajectories
/vis/scene/add/hits
```

How do I use DAWN from within Geant4?

```
/vis/open DAWNFILE
```

How can I make cuts using DAWN?

dawncut

What other visualization systems are available in Geant4?

VRML
OpenInventor (Coin3D)
RayTracerX
RayTracer

What external visualization options are available?

GDML -> python -> ROOT
WIRED

What is WIRED?

[WIRED Homepage](#)

How do I use WIRED?

[WIRED tutorial](#)

LCIO

What is LCIO?

LCIO stands for Linear Collider IO. It is the standard data format for exchanging event data between different ILC software, including simulators and reconstruction and analysis tools.

What languages does LCIO support?

- Java
- C++
- Python
- FORTRAN

What simulation tools support LCIO?

Nearly all of the simulation tools support LCIO, including the following tools.

- SLIC
- Mokka
- JUPITER
- BRAHMS

SLIC is unique in that it can read LCIO files as a physics event source.

What reconstruction and analysis tools support LCIO?

The [MARLIN](#) tool and its associated [MarlinReco](#) processors all support the LCIO format.

The [org.lcsim](#) Java framework includes its own implementation of LCIO in the package *org.lcsim.util.lcio*. This is compatible with the Java implementation included in the LCIO CVS project.

If I don't want to generate files myself, are there LCIO files available that I can use to get started with my analysis?

A [SLAC](#) ftp site has LCIO files available, mostly generated using SLIC. Follow the [ftp instructions](#) to find and download the files that you want to use.

How do I request new LCIO files?

Contact Norman Graf <ngraf@slac.stanford.edu> if you want some LCIO files generated by SLIC that are not available from the ftp site.

What tool can I use to look at LCIO output from the command line?

LCIO comes with the *dumpevent* tool that dumps an LCIO file's contents from the command-line. Typically, this would be used with a pager.

```
$LCIO/bin/dumpevent outfile.slcio 0 1 | less
```

The above command dumps the direct event of outfile.slcio file starting at run number 0.

What tool can I use to look at LCIO output from within JAS3?

The LCSim Event Browser can be used to look at LCIO files from within JAS3 using org.lcsim.

The [LCSim Event Browser instructions](#) show how to use this tool.

How can I create detailed diagnostic histograms of LCIO event data?

The SlicDiagnostics package creates AIDA files with detailed plots of LCIO event data, focused on the simulation classes: CalorimeterHit, TrackerHit, and MCParticle. Contact Jeremy McCormick <jeremym@slac.stanford.edu> if you would like more information about obtaining and using this package.

Where can I post my questions about LCIO?

[LCIO forum](#)

Where can I go for more information about LCIO?

[LCIO homepage](#)

Compact Detector Description

What is the Compact Detector Description?

The compact description provides an abbreviated format for describing a full detector and its subcomponents to the org.lcsim reconstruction and analysis subsystem.

The [compact detector description schema](#) specifies the structure of this format as an XML schema (XSD file).

How does the compact format work?

The package org.lcsim.geometry.compact transforms XML from the compact description into various targets. These include the LCDD language for the simulator and Java runtime objects for reconstruction and analysis.

Where is the standard location for compact XML files and their matching LCDD description?

Compact detectors are kept at this base url in zip files named with the detector tag.

```
http://www.lcsim.org/detectors
```

For instance, this is the location of the *sid00* detector's resource file.

```
http://www.lcsim.org/detectors/sid00.zip
```

The org.lcsim conditions system will use this URL automatically. Otherwise, a tool like *wget* can be used to download it manually.

Where is the list of available compact descriptions?

A list of available compact detectors is kept at this URL.

```
http://www.lcsim.org/detectors/taglist.txt
```

This list is updated periodically depending on the current detectors in LCDetectors.

How do I make org.lcsim find a full detector that is not in the standard area at lcsim.org?

Add an alias to `~/.lcsim/alias.properties` which points to your detector's resources.

A zip file.

```
mydet: file:///path/to/mydet.zip
```

Or the directory.

```
mydet: file:///path/to/mydet/
```

What is LCDetectors and how do I use it?

LCDetectors contains detectors in the compact description for use with org.lcsim.

Where can I find documentation on the detectors in LCDetectors?

[Detectors Wiki](#)

How do I convert compact detectors into LCDD using GeomConverter?

```
GeomConverter/bin/GeomConverter -o lcdd compact.xml mydet.lcdd
```

How do I define constants?

All attribute values are evaluated by the JDOM expression factory.

```
<constant name="myval" value="2.0 * cm"/>
```

After being defined, constants can be used from anywhere in the compact description. This is useful for reusing values in detector definitions.

What are the compact descriptions default units?

Unit	Default
length	mm
angles	radians
energy	GeV

What "magic" constants are expected by the compact reader?

```
<define>

  <!-- centimeters -->
  <constant name="cm" value="10"/>

  <!-- world -->
  <constant name="world_side" value="30000" />
  <constant name="world_x" value="world_side" />
  <constant name="world_y" value="world_side" />
  <constant name="world_z" value="world_side" />

  <!-- tracking region -->
  <constant name="tracking_region_radius" value="175.*cm"/>
  <constant name="tracking_region_zmax" value="282.*cm"/>

</define>
```

How do I create my own compact detector?

Defining a Detector

How do I define a cylindrical calorimeter?

Cylindrical calorimeter barrel.

```
<detector id="1" name="EMBarrel" type="CylindricalBarrelCalorimeter" readout="EcalBarrHits">
  <dimensions inner_r = "ecal_barrel_inner_r" outer_z = "ecal_barrel_outer_z" />
  <layer repeat="30">
    <slice material = "Tungsten" thickness = "ecal_absorber_thickness" />
    <slice material = "Silicon" thickness = "ecal_sensor_thickness" />
  </layer>
</detector>
```

Cylindrical endcap calorimeter.

```
<detector id="2" name="EMEndcap" reflect="true" type="CylindricalEndcapCalorimeter" readout="EcalEndcapHits">
  <dimensions inner_r = "ecal_endcap_inner_r" inner_z = "ecal_endcap_inner_z" outer_r = "ecal_endcap_outer_r" />
  <layer repeat="30">
    <slice material = "Tungsten" thickness = "ecal_absorber_thickness" />
    <slice material = "Silicon" thickness = "ecal_sensor_thickness" />
  </layer>
</detector>
```

How do I define a polyhedra calorimeter?

Polyhedra calorimeter barrel.

```
<detector id="3" name="EMBarrel" type="PolyhedraBarrelCalorimeter" readout="EcalBarrHits">
  <dimensions z="ecal_barrel_full_z" numsides="8" rmin="ecal_barrel_rmin"/>
  <layer repeat="30">
    <slice material="Tungsten" thickness="ecal_absorber_thickness"/>
    <slice material="Silicon" thickness="ecal_sensor_thickness" sensitive="yes"/>
  </layer>
</detector>
```

Polyhedra endcap calorimeter.

```
<detector id="4" name="EMEndcap" reflect="true" type="PolyhedraEndcapCalorimeter" readout="EcalEndcapHits">
  <dimensions numsides="8" rmin="ecal_endcap_rmin" rmax="ecal_endcap_rmax" zmin="ecal_endcap_zmin"/>
  <layer repeat="30">
    <slice material="Tungsten" thickness="ecal_absorber_thickness"/>
    <slice material="Silicon" thickness="ecal_sensor_thickness" sensitive="yes"/>
  </layer>
</detector>
```

How do I define calorimeter readout?

GridXYZ segmentation.

```
<readout name="HcalBarrHits">
  <segmentation type="GridXYZ" gridSizeX="10.0" gridSizeY="10.0"/>
  <id>layer:7,system:6,barrel:3,x:32:-16,y:-16</id>
</readout>
```

Nonprojective cylinder.

```
<readout name="EcalBarrHits">
  <segmentation type="NonprojectiveCylinder" gridSizePhi="0.35*cm" gridSizeZ="0.35*cm" />
  <id>system:8,layer:8,barrel:3,phi:32:16,z:-16</id>
</readout>
```

Projective Z plane (projective endcap).

```
<readout name="HcalEndcapHits">
  <segmentation type="ProjectiveZPlane" thetaBins="600" phiBins="1200"/>
  <id>layer:7,system:6,barrel:3,theta:32:11,phi:11</id>
</readout>
```

Projective cylinder.

```
<readout name="HcalBarrHits">
  <segmentation type="ProjectiveCylinder" thetaBins="600" phiBins="1200"/>
  <id>layer:7,system:6,barrel:3,theta:32:11,phi:11</id>
</readout>
```

How do I define a silicon tracker?

Tracker barrel.

```
<detector id="1" name="TrackerBarrel" type="MultiLayerTracker" readout="TkrBarrHits">
  <layer id="1" inner_r = "20.000*cm" outer_z = "26.7*cm">
    <slice material = "Silicon" thickness = "0.00048*cm" />
    <!-- N layers can go here -->
  </layer>
```

Tracker endcap.

```
<detector id="14" name="TrackerEndcap" type="DiskTracker" reflect="true" readout="TkrEndcapHits">
  <layer id="1" inner_r = "4.0*cm" inner_z = "30.0*cm" outer_r = "25.000*cm">
    <slice material = "Silicon" thickness = "0.03*cm" sensitive = "yes" />
  </layer>
  <!-- N layers can go here -->
</detector>
```

How do I define a TPC?

```
<detector id="2" name="TPC" type="TPC" readout="TPCHits">
  <dimensions inner_r = "36.2*cm" outer_z = "250.0*cm" />
  <layer repeat="200">
    <slice material="P10" thickness="tpc_gas_thickness" sensitive="yes" />
  </layer>
</detector>
```

How do I define a tracker readout?

Unlike calorimeters, tracker hits are written out "in the raw" without any digitization (or binning), so no segmentation is used at the simulation level.

```
<readout name="TkrBarrHits">
  <id>layer:10,system:6,barrel:3</id>
</readout>
```

How do I define a simple magnetic field?

```
<fields>
  <field type="Solenoid" name="GlobalSolenoid"
    inner_field="5.0"
    outer_field="-0.6"
    zmax="1000"
    outer_radius="144*cm+(2+1)*34*cm"/>
</fields>
```

How do I define a magnetic field map (RZB)?

The org.lcsim package can parse a field map with a line-oriented format. This gives the magnetic field strength at a given radius (r) and z.

```
r z Br Bz
```

Reference a field map from the compact description.

```
<fields>
  <field type="RZFieldMap" name="RZFieldMapTest"
    gridSizeZ="10.0"
    gridSizeR="10.0"
    numBinsZ="64"
    numBinsR="67"
    url="http://www.lcsim.org/test/solenoid_5tesla.dat"/>
</fields>
```

How do I define a new material?

```
<material name="ArgonGas">
  <D type="density" value="0.00178" unit="g/cm3"/>
  <composite n="1" ref="Ar" />
</material>

<material name="MethaneGas">
  <D type="density" value="0.000717" unit="g/cm3"/>
  <composite n="1" ref="C" />
  <composite n="4" ref="H" />
</material>

<material name="P10">
  <D type="density" value="0.00178" unit="g/cm3"/>
  <fraction n=".9" ref="ArgonGas" />
  <fraction n=".1" ref="MethaneGas" />
</material>
```

How do I set physics limits?

```
<limits>
  <limitset name="MyLimits">
    <limit name="step_length_max" particles="*" value="1.0" unit="mm" />
  </limitset>
</limits>
```

Reference from a slice element within layer.

```
<slice limits="MyLimits" ... />
```

How do I set the range cut?

```
<regions>
  <region name="MyRegion" store_secondaries="true" cut="0.1" lunit="mm" threshold="0.001" eunit="GeV" />
</regions>
```

Reference from a slice element within layer.

```
<slice limits="MyRegion" ... />
```

How do I add detector conditions such as sampling fractions?

These can be added to the LCDetectors directory for a detector tag. See existing detectors such as sid00 for examples.

How do I create my own subdetector type?

- define Java type in org.lcsim.geometry.subdetector
- define LCDD convertor in org.lcsim.geometry.compact.converter.lcdd

LCPhys

What is LCPhys?

LCPhys is a [Geant4 Physics List](#) tailored for linear collider detector simulation.

What models and physics processes does LCPhys use?

See the [Linear Collider Physics List Description](#) for details about the models and physics processes.

Who is the maintainer of LCPhys?

Dennis Wright, who is also the current Hadronics Coordinator of the Geant4 project, maintains the LC Physics List.

What simulation packages use LCPhys?

LCPhys is used by all three of the main ILC simulation packages, [SLIC](#), [Mokka](#), and JUPITER. SLIC uses LCPhys as its default physics list.

Are there validation plots available of LCPhys output?

There are currently some plots of [Thin Target Tests](#) available.

TODO: More LCPhys validation plots.

org.lcsim

What is org.lcsim?

The org.lcsim framework is a Java-based reconstruction and analysis package which has a plugin to [JAS3](#). JAS3 provides a number of integrated tools such as the [WIRED event display](#) and an [AIDA](#)-compliant plotter.

What tools do I need to use org.lcsim?

Java	1.5
Maven	1.0
JAS3	0.8.3 (optional)
Netbeans	5.0 (optional)

How do I install org.lcsim?

Developers can use [these instructions for building lcsim](#).

Users who just wish to use the Event Browser and Event Display can simply [install the JAS3 plugins](#).

What tools are provided by org.lcsim?

- physics utilities
 - stdhep reader

- 3, 4-vector utilities
 - space geometry
 - diagnostic generator
 - Jet finder, event shape utilities
- conditions framework
 - read detector constants from zip files
 - define new detectors with zip files on the web
 - local caching
 - reading compact geometry files
- driver framework
- fast MC
 - parameterized track and cluster smearing
 - smearing constants from conditions system
 - produces ReconstructedParticles
- IO framework
- event access (EventHeader)
- event display interface (WIRED)
- digitization
 - Digisim calorimeter digi
 - tracker, vertex digitization
- clustering
 - cheater
 - cone
 - nearest neighbor (NN)
 - minimal spanning tree (MST)
 - MIP
- tracking
 - SLDWeightMatrix, TRF, Garfield
- vertexing
 - ZVTop4
- analysis examples
 - cluster diagnostics
 - SlicDiagnostics
 - PFA
 - ClusterID
 - ParticleID
- AIDA plotting interface

How do I get started with writing my own analysis code in org.lcsim?

Follow the [lcsim Developer Tutorial](#).

How can I contribute code to org.lcsim?

Where can I go for more information on org.lcsim?

- [org.lcsim Homepage](#)
- [org.lcsim Wiki](#)

How do I use the contrib area?

Event Generation

What tools are used for ILC event generation?

How do I use Pandora-Pythia?

How do I use HERWIG?

How do I use WHIZARD?

Grid

What Grids are available for ILC users?

[ILC Grid at DESY](#)

What virtual organizations (VO) are associated with the ILC grid?

ilc
calice

Which tools are available on the Grid?

- Mokka

How do I get a Grid certificate?

SLAC users should get a PPDG-sponsored certificate.

[registration form](#)

Doug Olson <dlolson@lbl.gov> is responsible for the PPDG registration authority.

[request ILC VO membership at DESY](#)

[create a loadable Grid certificate](#)

What tools do I need to install to use the Grid?

How do I download files from the Grid?

Use the *lcg-cp* command.

```
lcg-cp --vo ilc lfn:{LFNfromDB} file:{absolute path}
```

How can I run a simulation job on the Grid?

How can I run an analysis job on the Grid?

How do I use the LCG2 tools at CERN?

[lxplus LCG2 instructions](#)

Where can I go for more information about the ILC Grid?

<http://grid.desy.de/>

<http://grid.desy.de/certs/>

<http://grid.desy.de/users/>

<http://grid.desy.de/install/DESY-VO.html>

<http://cic.in2p3.fr/>

<http://cern.ch/lcg/>

<http://www.eu-egee.org/>