SIMDET V4 Development - Users Guide

Contents

1 Preamble

Since the release of SIMDET V4.01 further developments are included.
A new section **"Modifications and Extensions"** describes the additional features.
The code development is based on CVS and is done in the Module **Simdet**.

---

2 Objectives

The program **SIMDET** simulates the response of the LC detector for e^+e^- physics as described in the Technical Design Report [1] in a fast and flexible manner. The main objectives of the program are threefold:

- It provides physics studies groups with a tool for evaluation of the physics potential of a LC.
- It treats the detector responds in a realistic matter using a parametrisation of results from the *ab initio* Monte Carlo program `BRAHMS`.
- It is fast and provides a simple output structure, so that physics studies can be done in a straight forward and decentralized way.

The basic components of the detector are assumed to be:

- a **vertex detector**
- a **tracker system** with main tracker, forward tracker and intermediate tracker with forward disks
- an **electromagnetic calorimeter**
- a **hadronic calorimeter**
- a **low-angle tagger** (instrumented mask) and
- a **low-angle luminosity calorimeter**.

The program applies Gaussian smearing to charged particle momenta and impact parameters with resolutions obtained from the ab initio Monte Carlo program **BRAHMS**. The calorimetric response is treated in a way also using a parametrisation of results from the program **BRAHMS**. Pattern recognition is emulated by means of cross reference tables between generated particles and detector response. An energy flow algorithm defines the output of the program. If wanted, the covariance matrix and dE/dx information can be added.

SIMDET offers for event input either to use a built in interface to **PYTHIA** for internal event generation or to read events generated by other programs and written by subroutine **LCWRITE**. In the case of using **PYTHIA**, beamstrahlung can be switched on. The history of the event can optionally enabled.

After simulation and for the purpose of physics analysis, particles are classified as

- **electrons**
- **photons**
- **muons**
- **charged hadrons**, interpreted as pions
- **neutral hadrons**, interpreted as long-lived kaons
- **clusters** of unresolved particles.

[1] F. Richard, J.R. Schneider, D. Trines and A. Wagner Edts., TESLA: The Superconducting Electron-Positron linear Collider with an Integrated X-Ray Laser Laboratory. Technical Design Report, Part IV A Detector for TESLA, DESY 2001-011 and ECFA 2001-209 (2001)

Tracking efficiencies, charge mismeasurement and threshold energies for detector response are also taken into account either by default or with user adjustable values. Low energy muons are interpreted as pions.

The output of the program can be directed to a file using the standard output structure or a restricted form. Also, output of 'reconstructed' events can be completely suppressed or, if requested, the best estimates for the energy flow objects are stored such that the physics analysis package **VECSUB** can be directly used.

The program runs under the control of free format data cards with the possibility of overriding the default options.

A lot of changes and improvements were made with respect to previous versions of **SIMDET**; hence the old versions should be considered as obsolete. **This document describes SIMDET V4 Development**

The code of **SIMDET V4.01** is found in [2].

3 Usage notes

SIMDET has a built in an interface to **PYTHIA** for internal event generation via subroutine **SIPYTH**. Alternatively, events generated by other programs can be used for detector simulation; subroutine **SIEVTI** provides an input interface to events written by subroutine **LCWRITE**. If the structure of the generated events does not coincide with common **/PYJETS/** of **PYTHIA**, subroutine **SIPREP** should fill the arrays **K_PY(I,...)**, **P_PY(I,...)** and **V_PY(I,...)**.

To allow for initial state radiation and beamstrahlung during event generation is in the responsibility of the event generator. Subroutine **SIPYTH** accounts for both radiations if wanted.

In all cases, **NEVENT** specifies the number of events to be processed.

The detector parameters used are defined in subroutine **SIDETR**. The naming convention is self explanatory.

After selection of all particles which might give a response in the detector (subroutine **SIPART**), charged particle tracking, impact parameter estimations and the covariance matrix are obtained from the subroutine **SITRAK**, subroutine **SICDAS** and subroutine **SICOVM**, respectively.

Detector resolution parameters for tracks are provided by the ab initio Monte Carlo program **BRAHMS** taking into account the complete tracking system.

Two options exist for the vertex detector (subroutine **SIDCAS**):

1. the APS version, based on 1.5 cm beam pipe radius
2. the CCD version, based on 1.5 cm beam pipe radius

and only the CCD version allows for the covariance matrix at present.

The calorimetric response is also based on detector studies using the *ab initio* Monte Carlo simulation package **BRAHMS**. The energy distributions of electrons and charged pions in adjacent cells were fitted and the parameters obtained are used for appropriate energy deposit simulations in the electromagnetic and hadronic calorimeters (subroutine **SILEGO** and subroutine **SIDEPO**). Afterwards, a cluster search algorithm (with some idealized assumptions) provides clusters (subroutine **SICLUS_I**) in the ECAL as well as the HCAL. Finally, an energy flow algorithm (subroutine **SIFLOW_I**) joins tracker and calorimeter informations such that as many single particles as possible are resolved. At this stage, the best estimate for charged particles is simply taken from the tracker, except for high energy electrons and positrons. Here, a proper weighting scheme of their tracker and calorimeter energies is applied (subroutine **SIWGHT**). Particle identification on the basis of shower shapes and matching between tracker and calorimeter information in space and momentum is emulated. Finally, best estimates of the energy flow objects are established.

In the very forward direction an instrumented mask, the low-angle tagger (subroutine **SILOWT**), and a low-angle luminosity calorimeter (subroutine **SILCAL**) are included as default devices. Their preliminary responses from **BRAHMS** simulation are included in **SIMDET**.

Particles not entering the calorimeters, e.g. low-pt tracks, are rescued in subroutine **SIRSCU** and added to the list of energy flow objects in subroutine **SISTOR**.

In addition, **SIMDET** allows for particle identification from specific energy losses dE/dx within the TPC (steered by subroutine **SIDEDX**).

[2] **/afs/ifh.de/group/lc/detector_simulation/pro/SIMDET.**

Background events of the type $\gamma\gamma \rightarrow hadrons$ (and later also of other sources) can be overlayed to each physical event using the package **hades** [3]. The data card **IBKG** enables this option provided the name of the background file (data card **BKGF**) and the average number of background events to be overlayed (data card **NBKG**) are given by the user. Also the study of only background events is possible. Formatted background files resulting from the package **GUINEA_PIG** exist for the **TESLA** collider at cms energies 170, 360, 500 and 800 GeV as well as for two **CLIC** options: the nominal **CLIC** parameter option with a luminosity $1.05 \cdot 10^{35} cm^{-2} sec^{-1}$ at 3 TeV and an optimized version with lower beamstrahlung and a luminosity of $0.4 \cdot 10^{35} cm^{-2} sec^{-1}$. It is the responsibility of the user to transform these files to a local binary file using the program **hades_import**.

The **CLIC** option of **SIMDET** can be enabled by the data card **CLIC**. If it equals 1, the process selected in subroutine **SIPYTH** is simulated at the predetermined cms energy, while **CLIC 2** allows for beamstrahlung effects if a luminosity spectrum file is provided on the data card **CLCF**. This option reads the electron and positron energies as obtained from **GUINEA_PIG** for a nominal beam energy of 1500 GeV. If a different energy is desired, the simulated e^+e^- energies are rescaled. In the case of invoking a luminosity spectrum file, note that the file name extension '.info' respectively '.ep' should not be given on the data card **CLCF**. For more details we refer to [4].

The history of the events can be monitored either completely or in a restricted form. If the data card **HISTORY** is enabled, the complete event history is stored as described in the **generated particle record**, otherwise input quantities of only stable particles are kept, with a flag of being accepted by the detector or not.

Booking of histograms is provided in subroutine **SIBOOK**. A few standard histograms allow to monitor the detector response. They can be optionally filled and switched off or on by the data card **PLOT**.

The simulation finishes by providing all measured objects in the proposed common output structure in subroutine **SISTOR**.

If requested, the program allows to write (unformatted/formatted) all objects in either the standard structure or in a restricted structure (best estimates only) to an external file in subroutine **SIWDST** so that analysis can follow externally. It also offers to produce zipped output files (patchy use selection **GZIO** and data card **GZIO**) in order to save disk space. If this option is chosen, the extension **.gz** is added by the program to the file name. The required package is provided with the **SIMDET** code.

If no external file should be written, the array **VECP(I,K)** is optionally filled in subroutine **SIVECS** which allows to use directly the physics analysis package **VECSUB** (a product from SLAC/DELPHI). In this case, the components 1 to 7 of the array **VECP** involve the quantities as described in the **VECSUB-DELPHI** note [5], while the originally free locations 8, 9 and 10 of this array contain now the transverse and the longitudinal impact parameters, in units of sigma, and a simple numbering of the energy flow objects accepted.

Subroutine **SIFFRE** reads the set of data cards with the **FFREAD** package. Default settings in subroutine **SIINIT** and/or in subroutine **SIDETR** might be overwritten.

A user subroutine **SIUSER (IFLAG)** has been added. It is called during initialization (IFLAG=1), for each event before writing to output file (IFLAG=2) and during termination of the job (IFLAG=3).

The package exists as a **CVS** repository as well as **PATCHY/CMZ CAR** files. If the **PATCHY** distribution is chosen, the desired code generation is steered by 'use' selections. For the **CVS** repository the selections are given as arguments to the configure script.

SIMDET selects the standard simulation code, and with **NOSIMU** only all stable particles without any detector response are treated. This enables physics studies at the parton or generator level.

Further **PATCHY** 'use' options are:

CIRCE to allow for the beamstrahlung code,

COVMTX for the covariance matrix material,

GZIO to produce zipped output files,

IBKG to invoke additional code for background events and

[3] D. Schulte and **/afs/cern.ch/eng/tev_phys/user1/hades/src**
[4] M. Battaglia and **/afs/cern.ch/eng/tev_phys/public/simdet/v2.0**
[5] G. Wormser, DELPHI note, November 16, 1988; revised by P. Roudeau, September 11, 1989

CLIC for the CLIC linear collider option.

Please note that for HP platforms error handling routines are included (forced by the 'use' selection **HPUX**).

Please link the **CERNLIB** libraries, the **PYTHIA 6.1** library and, if selected, the enclosed **GZIO** library, as transparently illustrated by the installation script which can be used for different platforms (Linux, HP-UX, SunOS, OSF1).

A small utility package **SIANAL** has been added which can serve as a template for further analyses. It contains routines to open and read the **SIMDET** output file of the energy flow objects written formatted or unformatted. For both types also zipped files can be read. A zipped file is assumed if the file name has the extension '.gz'. The file name is given in the command line to run **SIANAL**, eg. `'sianal.bin simdet_v4.evt.gz'` .

4 Input/Output units:

Logical unit number	Default file name	Contents
6	simdetv_4.res	debugging information like the history of a few events and histograms
7	user-defined	input events generated by an external program
8	simdetv_4.dat	free format data card file
9	user-defined	input $\gamma\gamma$ background events generated by an external program
10	user-defined	CLIC luminosity spectrum file generated by an external program
11	user-defined	simulated and reconstructed objects for further analysis (for zipped output the extension '.gz' is added automatically)
12	simdetv_4.hist	PAW histogram file

Note that the names of input and reconstructed event files should be enclosed in single quotes, start with a **point** and end with a **blank** when specified. The length of all file names is restricted to 80 characters.

5 Event output structure (standard):

This describes the SIMDET V4.01 standard output structure.

The output file starts with a single file header of 10 words. Each event starts with the number of generated particles **NGENPA**, which is followed by a record of 13 words for each particle. According to the user's choice, the whole event history (**HIST 1**) or only the stable particles which might give a response in the detector are covered.

The number of energy flow objects, **NEFLOW**, then follows. For each object, blocks of records follow, describing status information, best estimates, generator information, charged particles, calorimeter clusters and muons from the muon system, in this order.

5.1 File header record:

Offset	Variable name	Data type	Meaning
1	INSTATE	INTEGER	ISR flag, PYTHIA default: 0 = no radiation, 1 = ISR & Beamstrahlung, 2 = ISR only, 3 = Beamstrahlung
2	SQRTS	REAL	nominal collider cms energy [GeV]
3	SIGMA	REAL	Cross section [fb], if provided
4	IDCLASS	INTEGER	Reaction identifier, user defined
5	NEVENT	INTEGER	Number of events to be processed
6	IFBEST	INTEGER	Flag for best estimates
7	IFFORM	INTEGER	Flag for formatted output
8	IFDEDX	INTEGER	Flag for dEdx
9	IFHIST	INTEGER	Flag for complete (restricted) event history
10	IFBKGR	INTEGER	Flag for background events

This record is written once per file.

5.2 Generated particle record:

Offset	Variable name	Data type	Meaning
1	STATUS	INTEGER	0 = no detector response, 1 = detector response
2	ID	INTEGER	particle code according to PYTHIA convention
3	line	INTEGER	line number of particle in the PYTHIA record
4	Px	REAL	x-component of momentum [GeV]
5	Py	REAL	y-component of momentum [GeV]
6	Pz	REAL	z-component of momentum [GeV]
7	E	REAL	Energy [GeV]
8	m	REAL	mass of particle [GeV]
9	Q	REAL	charge of particle
10	x	REAL	x-component of vertex [mm]
11	y	REAL	y-component of vertex [mm]
12	z	REAL	z-component of vertex [mm]
13	Time	REAL	time of production [mm/c]

This record is repeated **NGENPA** times.

5.3 Energy flow record 1: status of energy flow object

Offset	Variable name	Data type	Meaning
1	STATUS	INTEGER	< 0: invalid object, 1=charged object, 2=neutral object, 3=composite object
2	Type	INTEGER	particle code according to PYTHIA convention, 999 = cluster of un-resolvable neutral and charged particles
3	NGEN	INTEGER	number of generator particles contributing to the energy flow object
4	NTRK	INTEGER	number of charged particles contributing to the energy flow object
5	NCAL	INTEGER	number of clusters contributing to the energy flow object
6	NMUS	INTEGER	number of muons contributing to the energy flow object

This record and the following blocks are repeated **NEFLOW** times.

5.4 Energy flow record 2: best estimate for object's energy and direction

Offset	Variable name	Data type	Meaning
1	Px	REAL	x-component of momentum [GeV]
2	Py	REAL	y-component of momentum [GeV]
3	Pz	REAL	z-component of momentum [GeV]
4	E	REAL	Energy [GeV]
5	m	REAL	mass of particle [GeV]
6	Q	REAL	charge of particle

This record is written once per energy flow object.

5.5 Energy flow record 3: generator particle contributing to energy flow object

Offset	Variable name	Data type	Meaning
1	link	INTEGER	line number of particle in the PYTHIA record
2	Efrac	REAL	energy fraction

This record is repeated **NGEN** times.

5.6 Energy flow record 4: charged particle tracks that are part of the object

Offset	Variable name	Data type	Meaning
1	P	REAL	absolute value of momentum [GeV]
2	Theta	REAL	polar angle [radian]
3	Phi	REAL	azimuth angle [radian]
4	Q	REAL	charge of particle
5	Imp(R,phi)	REAL	transverse impact parameter [cm]
6	Imp(R,z)	REAL	longitudinal impact parameter [cm]
7	cov. matrix	REAL	xy-xy element
8	cov. matrix	REAL	xy-z element
9	cov. matrix	REAL	z-z element
10	cov. matrix	REAL	xy-theta element
11	cov. matrix	REAL	z-theta element
12	cov. matrix	REAL	theta-theta element
13	cov. matrix	REAL	xy-phi element
14	cov. matrix	REAL	z-phi element
15	cov. matrix	REAL	phi-theta element
16	cov. matrix	REAL	phi-phi element
17	cov. matrix	REAL	xy-1/p element
18	cov. matrix	REAL	z-1/p element
19	cov. matrix	REAL	theta-1/p element
20	cov. matrix	REAL	phi-1/p element
21	cov. matrix	REAL	1/p-1/p element
22	dEdx	REAL	signed electron probability
23	dEdx	REAL	signed muon probability
24	dEdx	REAL	signed pion probability
25	dEdx	REAL	signed kaon probability
26	dEdx	REAL	signed proton probability
27	dEdx	REAL	normalized electron dEdx value
28	dEdx	REAL	normalized muon dEdx value
29	dEdx	REAL	normalized pion dEdx value
30	dEdx	REAL	normalized kaon dEdx value
31	dEdx	REAL	normalized proton dEdx value

This record is repeated **NTRK** times.

5.7 Energy flow record 5: calorimeter cluster (ECAL/HCAL/LAT/LCAL) that are part of the object

Offset	Variable name	Data type	Meaning
1	E	REAL	energy [GeV]
2	Theta	REAL	polar angle [radian]
3	Phi	REAL	azimuth angle [radian]
4	Time	REAL	timing information
5	C(em)	REAL	probability of being consistent with an el.-mag. particle
6	E	REAL	energy [GeV]
7	Theta	REAL	polar angle [radian]
8	Phi	REAL	azimuth angle [radian]
9	Time	REAL	timing information
10	C(mip)	REAL	probability of being consistent with a minimum ionizing particle

This record is repeated **NCAL** times.

5.8 Energy flow record 6: muons from the muon system that are part of the object

Offset	Variable name	Data type	Meaning
1	E	REAL	energy [GeV]
2	Theta	REAL	polar angle [radian]
3	Phi	REAL	azimuth angle [radian]
4	Q	REAL	charge of particle
5	Time	REAL	timing information
6	C(punch)	REAL	probability of being consistent with an punch-through object

This record is repeated **NMUS** times.

6 Restricted output structure:

This describes the SIMDET V4.01 restricted output structure, mainly intended to save disc space for reconstructed events. The output file starts with a single file header of 10 words. Each event starts with the number of energy flow objects, **NEFLOW**. This is followed by a series of records for each energy flow object, which describe only the best estimate of the object's energy and direction, in analogy to energy flow record 2 from the full data format, supplemented by impact parameter information.

6.1 File header record:

Offset	Variable name	Data type	Meaning
1	INSTATE	INTEGER	ISR flag, PYTHIA default: 0 = no radiation, 1 = ISR & Beamstrahlung, 2 = ISR only, 3 = Beamstrahlung
2	SQRTS	REAL	nominal collider cms energy [GeV]
3	SIGMA	REAL	Cross section [fb], if provided
4	IDCLASS	INTEGER	Reaction identifier, user defined
5	NEVENT	INTEGER	Number of events to be processed
6	IFBEST	INTEGER	Flag for best estimates
7	IFFORM	INTEGER	Flag for formatted output
8	IFDEDX	INTEGER	Flag for dEdx information
9	IFHIST	INTEGER	Flag for complete (restricted) event history
10	IFBKGR	INTEGER	Flag for background events

This record is written once per file.

6.2 Energy flow record:

Offset	Variable name	Data type	Meaning
1	Px	REAL	best estimate for x-component of momentum [GeV]
2	Py	REAL	best estimate for y-component of momentum [GeV]
3	Pz	REAL	best estimate for z-component of momentum [GeV]
4	E	REAL	best estimate for energy [GeV]
5	m	REAL	mass of particle [GeV]
6	Q	REAL	charge of particle
7	Imp(R,phi)	REAL	transverse impact parameter, in units of sigma
8	Imp(R,z)	REAL	longitudinal impact parameter, in units of sigma
9	NTRK	INTEGER	number of charged particles contributing to the energy flow object
10	type	INTEGER	particle code according to PYTHIA convention, 999 = cluster of un-resolvable particles

In cases of NTRK > 1, the largest impact parameters are recorded. This record is repeated **NEFLOW** times.

7 Data cards:

This describes the SIMDET V4.01 free format data cards.

The initialized default values are given in parentheses. Note that file names should be enclosed in single quotes, start with a **point** and end with a **blank**. The length of all file names is restricted to **80 characters**.

Certain obvious logical restrictions apply to the proper combination of options. For example, option 'PYTH 1' does not allow for event reading from an external file.

Key	Variable	Data type	Meaning
NEVT	NEVENT	INTEGER	total number of events to process (D=10)
PYTH	IFPYTH	INTEGER	0 = no PYTHIA event generation
			1 = event generation by PYTHIA without beamstrahlung
			2 = event generation by PYTHIA with beamstrahlung
			-1 = only $\gamma\gamma$ background events (D=1)
EVTI	IFEVTI	INTEGER	1 = event reading from an external file written by sr LCWRITE,
			0 = no event reading (D=0)
GENF	LUNGENF	INTEGER	input file name of generated events, user defined
WDST	IFWDST	INTEGER	1 = event writing to an external file,
			0 = no event writing (D=1)
RECF	LUNRECF	INTEGER	output file name of processed events, user defined
FORM	IFFORM	INTEGER	1 = formatted output structure,
			0 = unformatted output structure (D=0)
GZIO	IFGZIO	INTEGER	1 = zipped output is enabled,
			0 = no zipped output (D=0)
IBKG	IFBKGR	INTEGER	1 = background event reading from an external file,
			0 = no background event reading (D=0)
NBKG	BKGEVT	REAL	average number of bkgr. events to be overlayed to each processed event (D=0.)
BKGF	LUNBKGF	INTEGER	input file name of background events, user defined
CLIC	IFCLIC	INTEGER	0 = CLIC option disabled
			1 = CLIC option enabled without lumi spectrum file
			2 = CLIC option enabled with lumi spectrum file (D=0)
CLCF	LUNCLCF	INTEGER	input file name of CLIC lumi spectrum file, user defined
BEST	IFBEST	INTEGER	1 = best estimates are enabled,
			0 = no best estimates (D=0)
VECS	IFVECS	INTEGER	1 = filling of the array VECP(I,K) for VECSUB only if IFBEST = 1
			0 = no filling of this array (D=0)
HIST	IFHIST	INTEGER	1 = history of the generated event to output file,
			0 = only stable generated particles to output file (D=0)
COVM	IFCOVM	INTEGER	1 = covariance matrix is enabled,
			0 = no covariance matrix (D=0)
IPCT	IFIPCT	INTEGER	1 = IP beam constraint enabled,
			0 = no IP beam constraint (D=0)
PLOT	IFPLOT	INTEGER	1 = detector response plots are enabled,
			0 = no detector response plots (D=0)

Key	Variable	Data type	Meaning
APS	IFAPS	INTEGER	1 = pixel detector is enabled, 0 = no pixel detector (D=0)
CCD	IFCCD	INTEGER	1 = CCD detector is enabled, 0 = no CCD detector (D=1)
LCAL	IFLCAL	INTEGER	1 = low-angle calorimeter is enabled, 0 = no low-angle calorimeter (D=1)
LAT	IFLAT	INTEGER	1 = low-angle tagger (instrumented mask) is enabled, 0 = no low-angle tagger (D=1)
DEDX	IFDEDX	INTEGER	1 = dEdx information enabled, 0 = no dEdx information (D=0) 2. parameter (real): step size [cm], (D=3.0) 3. parameter (real): two-particle separation distance in the (r,z) projection [cm], (D=1.0) 4. parameter (real): two-particle separation distance in the (r,phi) projection [cm], (D=0.22) 5. parameter (integer): cut-off parameter 0 = track comparison is stopped after 5 consecutive steps 1 = complete track comparison (D=0)
EFTR	TRKEFF	REAL	measurement track efficiency (D=0.99)
PTTR	PTMINP	REAL	min transverse momentum required [GeV], (D=0.10)
QMTR	QTRKMIS	REAL	charge misinterpretation probability (D=0.005)
EMPH	EMECAL	REAL	min energy required for photon detection [GeV],(D=0.200)
EPIM	EPIMIS	REAL	electron misidentification probability (D=0.002)
AEMR	AEMRES	REAL	first energy resolution parameter for ECAL (D=0.145)
BEMR	BEMRES	REAL	second (constant) energy resolution parameter for ECAL (D=0.015)
EMHA	EMINHA	REAL	min energy required for hadron detection in HCAL, [GeV] (D=0.500)
AHAR	AHARES	REAL	first energy resolution parameter for HCAL (D=0.554)
BHAR	BHARES	REAL	second (constant) energy resolution parameter for HCAL (D=0.166)
ELAT	EMINLAT	REAL	min energy required for particle detection in LAT [GeV] (D=5.0)
ALAT	ALATRES	REAL	first energy resolution parameter for LAT (D=0.10)
BLAT	BLATRES	REAL	second (constant) energy resolution parameter for LAT (D=0.01)
CLAT	CLATRES	REAL	theta angular resolution in LAT [rad] (D=0.04)
DLAT	DLATRES	REAL	phi angular resolution in LAT [rad] (D=0.262)
EMMU	EMUDEP	REAL	deposited muon energy in calorimeters [GeV], (D=3.8)
EMIS	EMUISO	REAL	min energy for isolated muons [GeV], (D=5.0)
XMUM	XMUMIS	REAL	muon misidentification probability (D=0.005)

8 Installation

Access to the latest update of SIMDET is obtainable via the dev branch on the Web page
"http://www-zeuthen.desy.de/linear_collider".

The code of the newest SIMDET development version is provided as zipped tar file from a "cvs checkout Simdet". One can get it via the CVSWEB access, see:
http://www-zeuthen.desy.de/lc_repository/detector_simulation/dev/SIMDET/readme.html .

Un-taring the tarball creates a Simdet directory (and subdirectories). The Simdet subdirectory itself contains a **configure** script which generates the required Makefiles for the installation. The configure script accepts several options which can be visited by the invocation "configure -h" .

Running **configure** and after that **gmake** the executables of simdet and sianal are created. Optionally one can run a **gmake test**. Since Revision 1.16 the executables and the results of **gmake test** are stored in the subdirectory siuser.

If an option is used by the data cards (e.g. zipped output of events) but not installed, Simdet will send a message to the standard output and exit. However, if Simdet is installed with gzio it is still possible to use the un-zipped output for events.

9 Modifications and Extensions

Access to the latest update of SIMDET is obtainable via the dev branch on the Web page
"http://www-zeuthen.desy.de/linear_collider".

The code of the SIMDET development/extensions is provided as zipped tar file from a "cvs checkout Simdet". Links to these tar files can be found on
http://www-zeuthen.desy.de/lc_repository/detector_simulation/dev/SIMDET/readme.html .

Untaring the code creates a Simdet directory (and subdirectories). The Simdet subdirectory itself contains a **configure** script which generates the required Makefiles for the installation. The configure script accepts several options which can be visited by the invocation "configure -h" .

If an option is used by the data cards (e.g. zipped output of events) but not installed, Simdet will send a message to the standard output and exit. However, if Simdet is installed with gzio it is still possible to use the un-zipped output for events.

What is new since the release of SIMDET V4.01:

- The histogram storage is now initialized by HROPEN and closed by HROUT, HREND (Hbook package). This enables the user to book, fill, and write additional histos or ntuples in sr siuser
(located in the ./Simdet/sianal/code_f subdirectory).
- The debug printing is reorganized: a new data card DEBUG is introduced with:
DEBU n1 n2
n1 - first event number for debugging
n2 - last event number for debugging
- The **"vecsub"** package has been adapted to PYTHIA 6.1 and is added to the module Simdet (see the option **"–with-vecs"** in configure). The package provides many useful utility routines which may be called from siuser.
- Simdet accepts now generator input in StdHep format (eg. from the Whizard generator package). Any StdHep generator input file needs the extension **".stdhep"** and the corresponding data cards should be:

Key	Variable	Data type	Meaning
PYTH	IFPYTH	INTEGER	0 = no PYTHIA event generation
EVTI	IFEVTI	INTEGER	1 = event reading from an external file
GENF	LUNGENF	TEXT	input file name with the extension **.stdhep**

- The Fortran interface to the zlib (zipped I/O) has been changed to use several streams for I/O based on logical unit numbers. This enables the user to write his own output to a user defined logical unit number (must be > 12) as zipped file. How the interface can be used is described in the following section (as an example may serve the Fortran code in the subdirectory ./Simdet/sianal/code_f: sianal.F and anopen.F).

10 Description of the zlib Fortran Interface

SUBROUTINE GZOPEN (LUN, MODE, FILNAM, ISTAT)		
Opens a file for zipped IO		
	argument	meaning
	LUN	logical unit number used for this file
	MODE	string selecting IO mode ('r' or 'w')
	FILNAM	name of the file (including the null-terminator)
	ISTAT	status, =zero if success, #0 otherwise
Remarks:		
	LUN	in total 10 different LUN's can be defined hereby
	FILNAM	Extension '.gz' is checked

SUBROUTINE GZCLOSE (LUN, ISTAT)		
Closes a file used for zipped IO		
	argument	meaning
	LUN	logical unit number used for this file
	ISTAT	status, =zero if success, #0 otherwise

SUBROUTINE GZREAD (LUN, CHBUF, NBBUF, ISTAT)		
Reads a line from a zipped ASCII file		
	argument	meaning
	LUN	logical unit number used for this file
	CHBUF	array for storing unzipped info
	NBBUF	record size, number of bytes of array CHBUF
	ISTAT	status, = number of bytes read, =zero if EOF

SUBROUTINE GZBREAD (LUN, CHBUF, NBBUF, ISTAT)		
Reads a line from a zipped binary file		
	argument	meaning
	LUN	logical unit number used for this file
	CHBUF	array for storing unzipped info
	NBBUF	number of bytes (unzipped) to read
	ISTAT	status, =NBBUF if ok., =zero if EOF

GZWRITE (LUN, OUTLINE)		
Writes a line to a zipped ascii file		
	argument	meaning
	LUN	logical unit number for writing
	OUTLINE	character string to write
Remarks:		
	OUTLINE	written up to the last non-blank character

GZBWRITE (LUN, BUFFER, NBYTES)		
Writes the binary content in array BUFFER to a zipped file		
	argument	meaning
	LUN	logical unit number for writing
	BUFFER	array of binary content
	NBYTES	Number of Bytes to write

11 Final Remark

If you have questions, comments or suggestions, please do not hesitate to contact H. Jürgen Schreiber (email: schreibe@ifh.de) or Harald Vogt ((email: harald.vogt@desy.de).