

Contributing to ILC Software Projects

Requesting Bug Fixes and Improvements

The [ILC Jira](#) tracks bugs, feature requests, and possible improvements to existing software packages. Everyone is encouraged to add an item into the Jira when a bug is found in ILC software.

These Jira projects are specific to ILC simulations.

- [org.lcsim](#) - Java reconstruction and analysis package
- [Geometry Converter](#) - Java geometry conversion for org.lcsim
- [LCIO](#) - Linear Collider object model and persistency framework in C++, Java, and FORTRAN
- [SLIC](#) - C++ simulator package in Geant4 framework
- [LCDD](#) - C++ geometry description in Geant4 framework using [GDML](#)

Several other software packages are of more general interest.

- [JAS3](#) - Java-based HEP integrated development environment
- [AIDA](#) - plotting interfaces
- [JAIDA](#) - Java plotting implementation
- [WIRED](#) - event display using [HepRep](#)

Becoming a Developer

Before making any changes to ILC software, you need an account at the SLAC CVS.



CVS Account

Email tony_johnson@slac.stanford.edu to request a SLAC CVS account.

Follow the [CVS Instructions](#) to checkout the software projects of interest.

The reconstruction software is based on the [Java language](#), which you'll need to learn if you want to contribute. [Sun's Java Tutorial Book](#) is an excellent place to start if you are new to this language.



Python, Too

Scripts can also be written in the [Python](#) language using the [Jython](#) interpreter.

Read over the [development tools](#) section to learn about the tools we use to build and development the software.

Work through the [Getting Started Tutorial](#) to learn about the basic concepts.

Lists of Project Ideas

A [list of specific issues](#) was compiled at the [ALCPG Simulation and Reconstruction Workshop](#).

[Jeremy McCormick](#) maintains a [list of project ideas](#). (Real bugs and feature requests go into the Jira.)

Some Example Projects

- Geometry System in Reconstruction
 - allow geometries to come from multiple input files
 - improve interoperability
 - access Mokka or JUPITER geometry data
 - ID decoding for detectors not run with SLIC
- Reconstruction
 - process LCIO files using both org.lcsim and MARLIN
- Simulation Software
 - support for putting GDML data in multiple input files
 - additional calorimeter readout types to the LCDD package or improve on existing readouts
 - CAD interface
 - use case - LAPP ECAL for SiD
- Simulation Validation
 - investigate and characterize effects of range cuts and physics limits on Geant4 simulation
 - comparison of different physics lists (preferably with real data, too)
 - single particle response of h0, in particular

- Event Generation
 - python wrapper to the different generators used at ILC, e.g. Pythia, Isajet, etc.
 - SUSY StdHep files for benchmarks
- Testbeam
 - use JAS3 and org.lcsim to create a data analysis application for the Calorimeter testbeam
 - do some analysis on real ECAL data
- AIDA (+ Plotting)
 - binary data format in order to store Ntuples
 - 3D plotting

Links

[A University Program of Accelerator and Detector Research for the International Linear Collider \(vol. III\)](#)