Key4hep release on /cvmfs/ilc.desy.de

Software and Analysis Meeting

Thomas Madlener

Apr 13, 2022

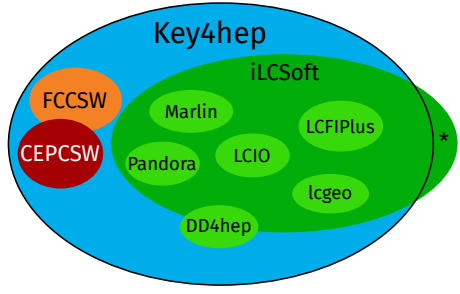
Key4hep in brief

- The Key4hep project aims to define a common software stack for all future collider projects
  - Re-use existing tools and software libraries where possible
  - Develop new things where necessary
  - Support all necessary workflows for doing physics studies, i.e. generation, simulation, reconstruction, analysis, ...
- **Collaboration of all major communities: ILC, CLIC, FCC-ee & FCC-hh, CEPC, (EIC), ...**
  - Regular (open) meetings: https://indico.cern.ch/category/11461/
- Part of the CERN Strategic R&D Programme on Technologies for Future Experiments and AIDAinnova WP12

Key4hep release contents

key4hep/key4hep-spack

- A “Key4hep release” is an installation of the `key4hep-stack` bundle package via spack
- “iLCSoft” within Key4hep is an installation of the `ilcsoft` bundle package via spack
  - `ilcsoft` is part of `key4hep-stack`
  - Contains everything that is usually in an iLCSoft release,
  - … and a bit more (e.g. EDM4hep and `k4MarlinWrapper`)
  - **Does not have an equivalent iLCSoft release!**
- The current release comes with
  - `gcc@11.2.0`
  - `python@3.9.10`
  - `root@6.26.00`
  - `geant4@11.0.1`
  - The latest tags for iLCSoft packages



*Some testbeam related packages are not (yet) available

Setup scripts

- Set up the complete Key4hep release:

```
source /cvmfs/ilc.desy.de/key4hep/setup.sh
```

*This is almost equivalent to `/cvmfs/sw.hsf.org/key4hep/setup.sh`

- Set up the iLCSoft part (a subset of `setup.sh`):

```
source /cvmfs/ilc.desy.de/key4hep/init_ilcsoft.sh
```

- These work on machines running CentOS7
- These are just wrapper scripts and **will always setup the latest available release!**
- For more stable results use dedicated setup scripts
- **We try to keep things working, but this is not yet stable for production!**

```
15:05:22 madlener@naf-ilc11:~$ source /cvmfs/ilc.desy.de/key4hep/setup.sh
Setting up the latest Key4hep release: 2022-04-06 (x86_64-centos7-gcc11.2.0-opt/o64eaqrx3j7fh6ofwbsqh4tmispz7nzu)
... To reproduce this exact setup at a later stage do
...
    source /cvmfs/ilc.desy.de/key4hep/spackages/key4hep-stack/2022-04-06/x86_64-centos7-gcc11.2.0-opt/o64eaqrx3j7fh6ofwbsqh4tmispz7nzu/setup.sh
...
... done.
```

Setup scripts

- Set up the complete Key4hep release:

```
source /cvmfs/ilc.desy.de/key4hep/setup.sh
```

*This is almost equivalent to `/cvmfs/sw.hsf.org/key4hep/setup.sh`

- Set up the iLCSoft part (a subset of `setup.sh`):

```
source /cvmfs/ilc.desy.de/key4hep/init_ilcsoft.sh
```

- These work on machines running CentOS7
- These are just wrapper scripts and **will always setup the latest available release!**
- For more stable results use dedicated setup scripts
- **We try to keep things working, but this is not yet stable for production!**

```
15:05:22 madlener@naf-ilc11:~$ source /cvmfs/ilc.desy.de/key4hep/setup.sh
Setting up the latest Key4hep release: 2022-04-06 (x86_64-centos7-gcc11.2.0-opt o64eaqrx3j7fh6ofwbsqh4tmispz7nzu
... To reproduce this exact setup at a later stage do
...
    source /cvmfs/ilc.desy.de/key4hep/spackages/key4hep-stack/2022-04-06/x86_64-centos7-gcc11.2.0-opt/o64eaqrx3j7fh6ofwbsqh4tmispz7nzu/setup.sh
...
... done.
```

tab tab tab

Differences to a "normal" iLCSoft environment

- For **running software the environment should be identical**
  - I.e. everything that works, e.g. with `v02-02-03/init_ilcsoft.sh` should also work with `key4hep/init_ilcsoft.sh`
  - If you discover something that works with the former but not with the latter, please let me/us know!
- There are some **minor differences when you want to build your packages**
  - In a “normal” iLCSoft environment you need to `cmake -C $ILCSOFT/ILCSoft.cmake <other-args>`
  - In a Key4hep based environment you only need to `cmake <other-args>`
  - The Key4hep environment also comes with Ninja build system (`cmake -G Ninja <other-args>`) but default `make` works as well

I need a package and it is not there yet

- For python packages you can use `pip install <package>`
  - Uses the python and pip of the environment
  - Uses installed packages in the environment
- If you want to build software on top of the stack you should be all set
  - Especially for packages using CMake
- If you want to use your own spack installation you can point to this one as `upstream` repository
- You can also open an issue at key4hep/key4hep-spack and ask for additional packages to be included in the stack
  - It might be necessary to formalize the build process in a *spack package* (python module)
- **If you find something that should work but doesn't let us know**

From iLCSoft to Key4hep

- iLCSoft is still the working horse for all ILD studies (and will be for some time)
- Migrating to Key4hep is a longer term goal
- On a very high level the environments serve the same purpose: enabling physics studies
- The major differences between the two are the experiment framework and the event data model

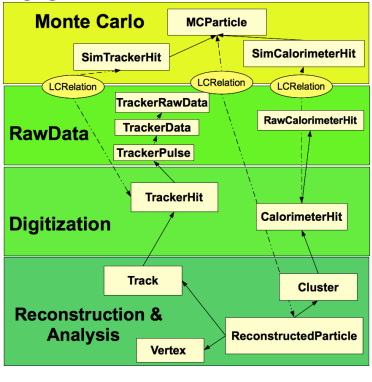
	iLCSoft	Key4hep
framework	Marlin	Gaudi
EDM	LCIO	EDM4hep

- In order to ensure a smooth transition between the two there is a way to “combine” the two worlds

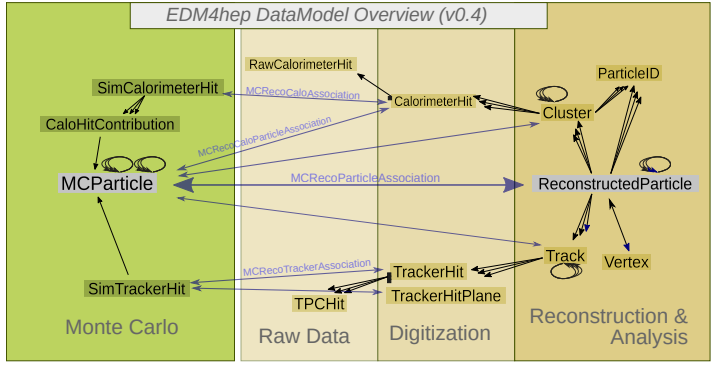
LCIO vs EDM4hep

A side-by-side comparison

LCIO



EDM4hep



- Since EDM4hep is based on LCIO the high-level structure is very similar
- Largest differences between the two are due to their implementations

Marlin vs Gaudi

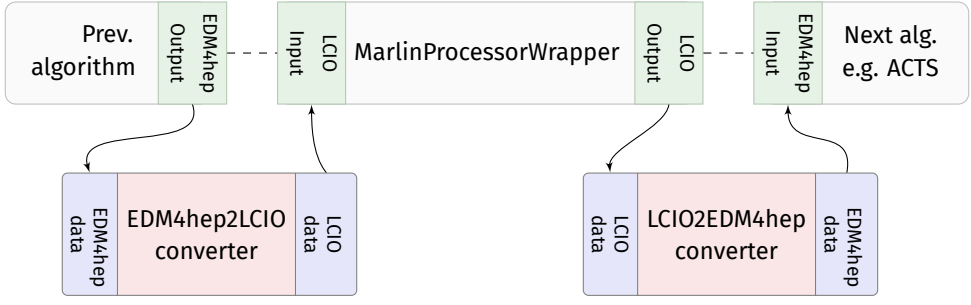
- Conceptually the two frameworks are very similar
  - Schedule different working units
  - Marshall data
- Most obvious differences in naming conventions
  - As always some differences emerge when looking at the details

	Marlin	Gaudi
language	c++	c++
working unit	Processor	Algorithm
config language	XML	Python
transient data format	LCIO	anything
set up function	`init`	`initialize`
work function	`processEvent`	`execute`
wrap up function	`end`	`finalize`

k4MarlinWrapper

Running Marlin processors in the Gaudi framework

- Wraps **Marlin processor** in a Gaudi algorithm and allows to **run them unchanged**
  - Can run a full ILD / CLIC reconstruction and analysis chain via Gaudi
- Converter script to turn Marlin XML steering files into Gaudi python option files
- Automatic, on-the-fly conversion between LCIO and EDM4hep
  - Allows to “mix and match” existing Marlin processors with Gaudi algorithms



CLIC reconstruction via Gaudi

- Main key4hep documentation: key4hep.web.cern.ch
  - Geared towards the official release at `/cvmfs/sw.hsf.org/key4hep`
  - Should work just the same with `/cvmfs/ilc.desy.de/key4hep`
- Also has an example for running the CLIC reconstruction in Key4hep [link]
  - Has the necessary steps to run via `Marlin`
  - Explains how to run the same chain via Gaudi (`k4run` is the python executable that sets up and runs Gaudi)
  - Also shows how to convert `xml` steering files to python option files
- This should be straight forward to apply to the ILD standard reconstruction as well
- **If something is missing (or wrong) please let us know!** (even if it just in the documentation)

Summary

- A new installation of the Key4hep software stack is available at

```
/cvmfs/ilc.desy.de/key4hep/setup.sh
```

- It comes with a complete installation of the iLCSoft stack
- This is currently aimed at users that want to take some first steps in the new Key4hep world
- You can run your favorite analysis in this environment via `Marlin`
- You can also try and run the same analysis via `Gaudi` with the use of the `k4MarlinWrapper`
- **We are happy about all feedback**