

# FairSHiP

## Software for SHiP



### Tutorial: change of schedule

There was a change in today's schedule, due to overlapping with the LIGO/VIRGO announcements.

New timetable:

- **Today: 15:00 to 16:30** (room 13-2-005)
  - Introduction (**Thomas**)
  - Introduction to HNL simulation and analysis (**Elena**)
  - How to implement detectors and material (**Annarita**)
- **Tomorrow: 8:00 to 9:00** (Filtration Plant)
  - (Real) hands-on session: creating a detector (**Iaroslava**)

Please attend the early-morning Friday session, it will be the most useful towards the preparation of the CDR!

# FairSHiP

## Software for SHiP



*Thomas Ruf (CERN)*

## Introduction

- **Basics**
- **Software Overview**
- **Structure**

## ■ Mailing list

- ◆ [ship-software@cern.ch](mailto:ship-software@cern.ch) (SHIP Collaboration mailing list dedicated to software )
- ◆ Archive: <https://groups.cern.ch/group/ship-software/default.aspx>

## ■ Web pages

- ◆ <http://ship.web.cern.ch/ship/FairShip/default.html>
- ◆ <https://twiki.cern.ch/twiki/bin/view/Ship/ShipSoftware>

## ■ Instructions for working at CERN

- ◆ <http://ship.web.cern.ch/ship/FairShip/computingAtCERN.html>

## ■ Software repositories

- ◆ <https://github.com/ShipSoft>
  - ▶ <https://github.com/ShipSoft/FairSoft>
  - ▶ <https://github.com/ShipSoft/FairRoot>
  - ▶ <https://github.com/ShipSoft/FairShip>
- ◆ Pre-installation on afs:
  - ▶ </afs/cern.ch/sw/ShipSoft>

## ■ Data repository

- ◆ EOS: </eos/ship/data>, quota 10TB

## ◆ Documentation:

<https://git.cern.ch/web/shipdocs.git/tree/HEAD:/Public/FairShip>

<https://trufship@git.cern.ch/repos/shipdocs/> (not updated since summer'15, sorry.)



# Software Overview

## ■ 3 Pillars

◆ <https://fairroot.gsi.de/>

◆ [September 2014 Tutorial](#)

C++



C++



C++

and Python for configuring of C++ objects and execution of non-CPU critical algorithms, flexible, user friendly!

## ■ GIT

- ◆ **Widely used distributed version control system**
- ◆ **Every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities**
- ◆ **Central hub:** <https://github.com/ShipSoft>
- ◆ **Main commands:**
  - ▶ `git clone https://github.com/ShipSoft/FairShip.git` : create local copy of FairShip
- ◆ **In FairShip directory**
  - ▶ `git pull` : fetch all updates from central hub
  - ▶ `git checkout c71af77179224b6d0f7393c69ab257df8b50722c` : specific version
  - ▶ If you want to know more: <https://github.com/ShipSoft/FairShip/wiki/Git-Tutorial-for-SHIP>

## ■ **Software development**

- ◆ For the moment, one person (me) doing commits and push
- ◆ If you have any changes, additions, etc., please present it in the ShipSoft meeting and send me the code.
- ◆ If this turns out to be not maintainable, people will have to learn how to work with Git in more details.

# Installation

## ■ Same procedure for FairSoft, FairRoot and FairShip

- ◆ `git clone https://github.com/ShipSoft/XXX.git`
- ◆ `cd XXX`
- ◆ `./configure.sh`
- ◆ Need two env variables for boot strapping: **SIMPATH** and **FAIRROOTPATH**

## ■ On Ixplus, profit from installed FairSoft, FairRoot:

- ◆ `export SHIPSOFT=/afs/cern.ch/ship/sw/ShipSoft`
- ◆ `export SIMPATH=${SHIPSOFT}/FairSoftInst`
- ◆ `export FAIRROOTPATH=${SHIPSOFT}/FairRootInst`

## ■ After installation of FairShip, to setup the environment, use:

- ◆ `source $FAIRSHIPRUN/config.sh` (\$FAIRSHIPRUN pointing to your FairShip/./FairShipRun)

## ■ Changing existing code, recompilation

- ◆ `cd $FAIRSHIPRUN ; make`

## ■ Adding new c++ code

- ◆ For advanced users: `cd $FAIRSHIPRUN ; cmake ; make`
- ◆ To be on safe side: `rm -rf $FAIRSHIPRUN ; ./configure.sh`

## ■ Geometry

### ◆ Subdetector directories and passive materials

- ▶ /nutaudet, /veto, /strawtubes, /ecal, /hcal, /muon, /passive

### ◆ Also contain

- ▶ Definition which volumes are sensitive
- ▶ What information to store for MC particles entering the volume, momentum, entry/exit points

## ■ Global data objects

### ◆ shipdata directory

- ▶ shipstack, work space for Geant
- ▶ ShipMCTrack, MC particle object of FairShip

## ■ MC Generators

### ◆ shipgen directory

### ◆ Implemented use cases:

- ▶ HNL signal from charm (beauty): HNLPythia8Generator
- ▶ Muon background: MuonBackGenerator
- ▶ Muon inelastic interactions: MuDISGenerator
- ▶ Neutrino inelastic interactions: GenieGenerator, NuageGenerator
- ▶ Cosmic background: CosmicsGenerator

## ■ Configuration

### ◆ python directory

- ▶ shipDet\_conf.py, DecaySelection.conf
- ▶ Also some other useful modules: shipunits.py, ShipStyle.py

### ◆ geometry directory

- ▶ Geometry parameters for ecal and hcal
- ▶ List of materials, media.geo

## ■ Execution

### ◆ macro directory

### ◆ Scripts to run simulation, reconstruction, analysis and eventdisplay

### ◆ Accept command line arguments for different use cases

- ▶ run\_simScript.py          simulation  
    Philosophy: one script for many use cases, instead of many scripts each for one use case.
- ▶ ShipReco.py                reconstruction
- ▶ ShipAna.py                template for analysis
- ▶ eventdisplay.py          visualization of detector geometry and event data

### ◆ genfit directory

- ▶ External package for track fitting, extrapolation of track states through magnetic field and material
- ▶ Tutorial by Sebastian Neubert, <https://indico.cern.ch/event/336469/session/0/contribution/5/attachments/658106/904773/genfitintro.pdf>



# Not covered today

## ■ Production and analysis on SkyGrid

- ◆ See presentation by Alexander Baranov (Sasha) November 2015
- ◆ Requires a git repository of your code if different from official version
- ◆ The command line to execute
- ◆ Send request to [skygrid-users@cern.ch](mailto:skygrid-users@cern.ch)

## ■ FairSoft/FairRoot/FairShip can easily be installed on your laptop/desktop

- ◆ With linux operating system
- ◆ Or using a Virtual Machine
  - ▶ <https://github.com/ShipSoft/FairShip/wiki/Linux:-Building-&-Running-FairShip-using-Virtual-Machine-container>
  - ▶ VMWare (my solution), <https://www.vmware.com/go/downloadplayer/>
- ◆ A local installation is preferred solution for running event display

# Not covered today, cont.

## ■ Event Display

- ◆ Very useful for getting an idea about detector geometry, how the particles traverse the setup
- ◆ For making PR pictures
- ◆ `python -i $FAIRSHIP/macro eventdisplay.py -f ship..._rec.root -g geofile...root`

