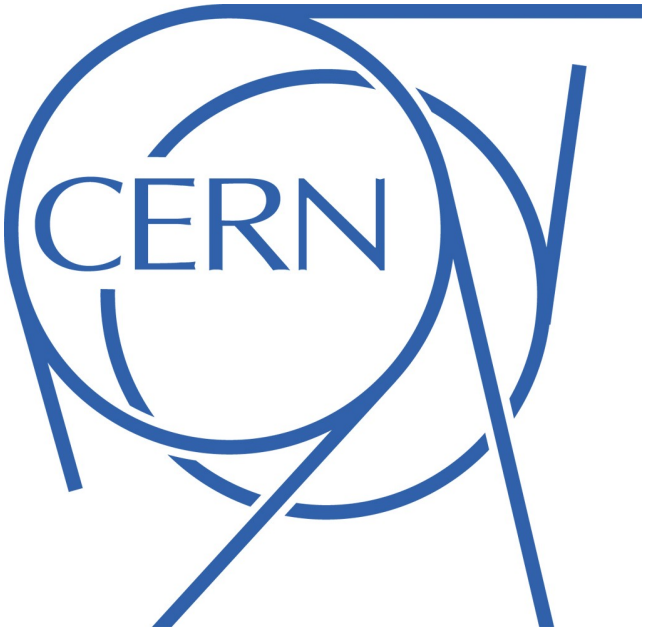
Development of the O2 prototype



Charis Kouzinopoulos
CERN
ALICE Offline week - Thursday, 20 November 2014

Summary of Developments

- The AliceO2 repository was created on the 8th of October
- The project is hosted on GitHub under AliceO2Group
- The code can be checked out at:

  https://github.com/AliceO2Group/AliceO2.git

- The project was constructed using the FairRoot project template as a starting point
- Part of the FairRoot distribution:

  https://github.com/FairRootGroup/FairRoot/tree/dev/templates/project_template

- The template comes with an example detector with sensitive and passive volumes (NewDetector), event generators, etc
- To initialize the project and replace all the generic names to user defined ones, use the *rename.sh* script:

  rename.sh <project name> <class prefix> <detector name to be implemented>

- A detailed description of the project template is available here:

  https://github.com/FairRootGroup/FairRoot/tree/dev#using-the-project-template

To build the AliceO2 project, the following steps can be used:

- Install the FairSoft 'external' packages (latest tag *jul14p3*):
  https://github.com/FairRootGroup/FairSoft/tree/jul14p3
  These include Boost, Geant, Pythia, Root, ZMQ, Protocol Buffers etc

- Set the variable SIMPATH to point to the FairSoft installation directory

- Install the *dev* branch of FairRoot:
  https://github.com/FairRootGroup/FairRoot/tree/dev

- Set the variable FAIRROOTPATH to the FairRoot installation directory

- Compile the *dev* branch of AliceO2:
  https://github.com/AliceO2Group/AliceO2/tree/dev

The development workflow used is based on Anar Manafov's proposal:

https://github.com/AnarManafov/GitWorkflow/blob/master/GitWorkflow.markdown

See also yesterday's detailed presentation by Dario Berzano

Purpose of the Git workflow is to:

- Keep the codebase clean (i.e., no back-and-forth merging)
- Have a master that always works
- Maintain a clean history without merge commits and other garbage
- Have multiple levels of protection against conflicts
- Have multiple possibilities to recover from errors/mistakes before changes come into the master

There are only two long-term branches: master and dev

The master branch:

- Contains only stable code
- It is *always* ready to build – always kept in a releasable state
- No development is performed directly to master
- Only administrators have write permissions on it
- No history changes are allowed on master
- All new patches are introduced in master only via "git merge --ff-only"

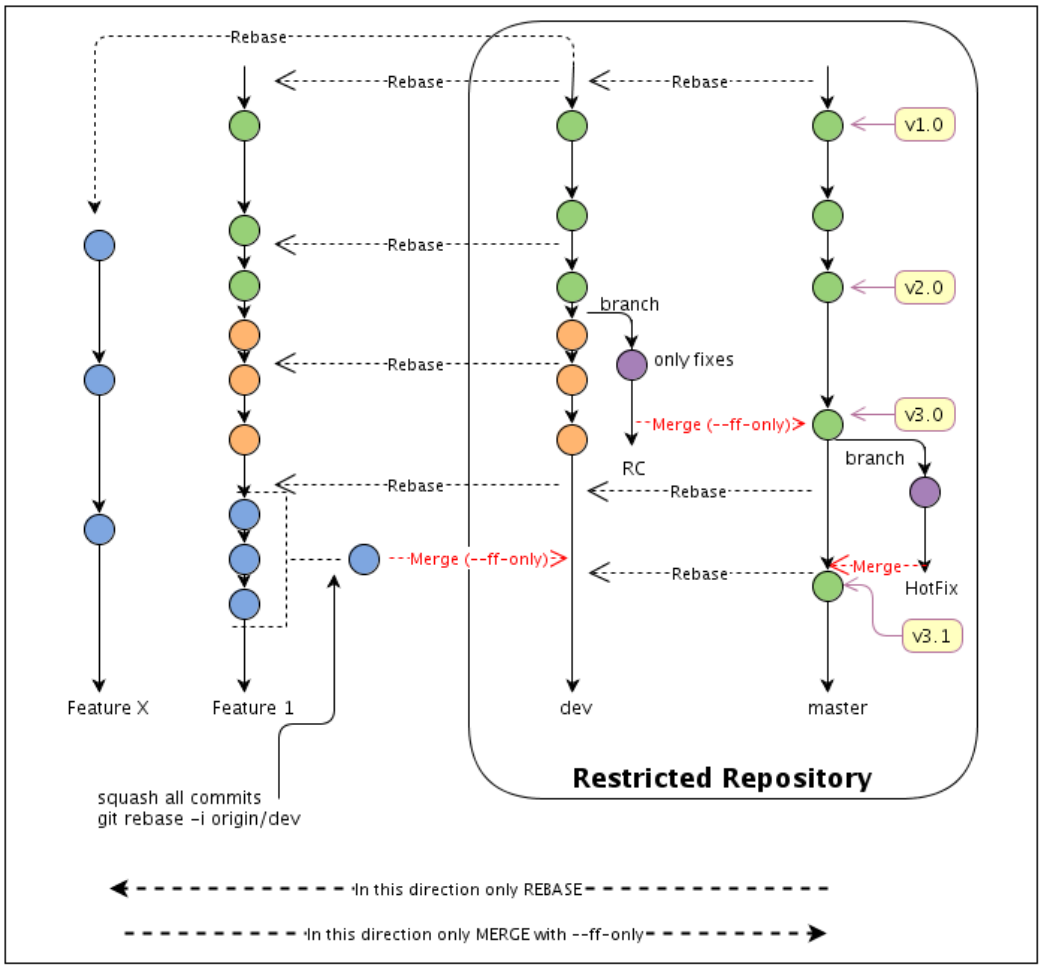
The dev branch:

- Is the current development branch
- Is inherited from the latest master
- It the place where developers code is merged into

From the developers perspective:

- The developers *fork* the repository and create a local copy

- Locally, one branch is used per feature/bug
- Multiple commits per feature/bug are *squashed* into one
- Before each code push, the local branch is *rebased* against dev
- The branch is merged to dev using fast forward to combine the commit history



To develop AliceO2, we are implementing the C++ Coding Guidelines and the C++ Naming & Formatting Rules as introduced by CWG2:

https://atelesca.web.cern.ch/atelesca/coding_guidelines/cppguide.xml

https://atelesca.web.cern.ch/atelesca/naming_formatting/cppguide.xml

Some examples:

- Names are descriptive and follow camel case convention: **AliceO2, ContainerFactory, getLayerParameters**
- Curly braces placement

  ```
  if (condition) {
  } else {
  }
  ```

- Use spaces instead of tabs, 2 space indentation, 120 characters per line etc

- To format the AliceO2 code according to the Formating Rules, we are using ClangFormat
- The ClangFormat configuration file, *.clang-format* is included in the AliceO2 tree
- To apply the Formating Rules to a given source file, execute:

  clang-format-3.5 -style=file -i SOURCEFILE

AllowShortLoopsOnASingleLine:	false
AlignTrailingComments	true
ColumnLimit	120
IndentWidth	2
SpacesInParentheses	false
UseTab	never

Important notes:

- Use of namespaces everywhere in the form of Project::Component::

  AliceO2::ITS::Detector::defineWrapperVolume
  AliceO2::Base::Detector::defineWrapperVolume

  The implementation of namespaces in the CINT intrepreter seems to be incomplete. It cannot distinguish between the following:

  AliceO2::ITS::Detector::defineWrapperVolume
  Detector::MaskToString

  ...as present in the dbase/dbValidation/Detector.h class of FairRoot

- Since with the new naming scheme, multiple headers with the same name can exist, it is essential to ensure their correct inclusion by using a strict naming for header guards in the form of: <PROJECT>_<PATH>_<FILE>_H_.

  #define ALICEO2_ITS_DETECTOR_H_
  #define ALICEO2_BASE_DETECTOR_H_

Important notes:

- Use of namespaces everywhere in the form of Project::Component::

  AliceO2::ITS::Detector::defineWrapperVolume
  AliceO2::Base::Detector::defineWrapperVolume

  The implementation of namespaces in the CINT intrepreter seems to be incomplete. It cannot distinguish between the following:

  AliceO2::ITS::Detector::defineWrapperVolume
  Detector::MaskToString

  ...as present in the dbase/dbValidation/Detector.h class of FairRoot

- Since with the new naming scheme, multiple headers with the same name can exist, it is essential to ensure their correct inclusion by using a strict naming for header guards in the form of: <PROJECT>_<PATH>_<FILE>_H_.

  #define ALICEO2_ITS_DETECTOR_H_
  #define ALICEO2_BASE_DETECTOR_H_

For the code documentation, we are using doxygen

The C++ Comments Guidelines of CWG2 can be found at:
https://atelesca.web.cern.ch/atelesca/comments_guidelines/cppguide.xml

Some examples:

To comment code (double slashes):
// …. text ….

To document a class/method (triple slashes):
/// …. text ….

To document data members:
private:
int mTotalNumberOfEntries; ///< Total number of entries

In classes using ROOT IO, for the data members excluded from IO:
double mBuffer; //!< Temporary buffer

The doxygen configuration file, *AliceO2Doxygen.conf*, is included in the AliceO2 tree

To create the doxygen documentation, execute:

doxygen AliceO2Doxygen.conf

Next step: automatically update and upload the documentation

AliceO2

Main Page | Modules | Namespaces | Classes | Files

AliceO2 Documentation

Alice O2 project software. Simulation and reconstraction software for the ALICE experiment at CERN based on ALFA and the FairRoot software.

Step by Step installation

1. Install FairSoft/AlFa

   we use here "alfa_src" as a directory name, you can change it to what ever you like

```
git clone  https://github.com/FairRootGroup/FairSoft.git  alfa_src
cd  alfa_src
./alfaconfig.sh
# 1) gcc (on Linux) 5) Clang (on OSX)
# 1) No Debug Info
# 2) Internet (install G4 files from internet)
# path: ~/AlFa
```

   To run the tests do:

```
cd alfa_src/FairRoot/build_for_alfa/
make test
```

2. Set several required shell variables, needed during the installation and running of the different software packages. Put these in your shell's rc file (~/.bashrc or ~/.cshrc). For bash:

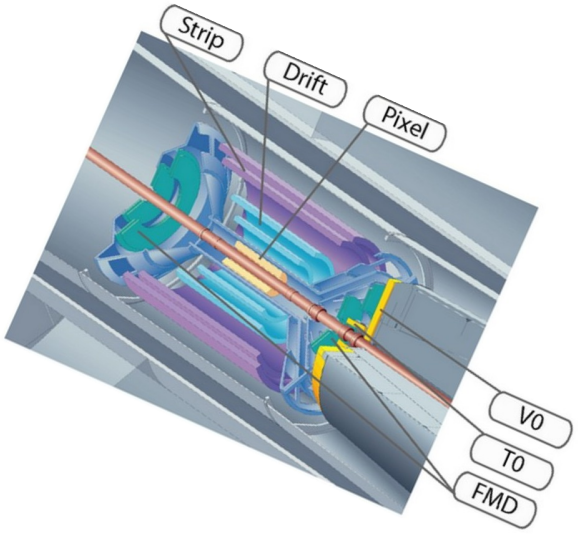
AliceO2 – Tree summary

The AliceO2 tree currently holds the following main modules:

- Devices: Data Transportation, FLP to EPN code, HLTWrapper code
- ITS: Simulation code for the ITS detector
- Base: Abstraction classes for the ITS detector
- Data: Particle stack for the transport simulation, storage of Monte Carlo tracks processed by the particle stack
- Field: Magnetic field classes
- Resources: Magnetic field maps

AliceO2 contains a port of the simulation code for the ITS detector

- The previous port was based on *ITS Upgrade v1* code of AliRoot tag *v5-05-64-AN*
- The current port was updated to *ITS Upgrade v1* code of AliRoot tag *vAN-20140922*
- There are no dependencies on AliRoot

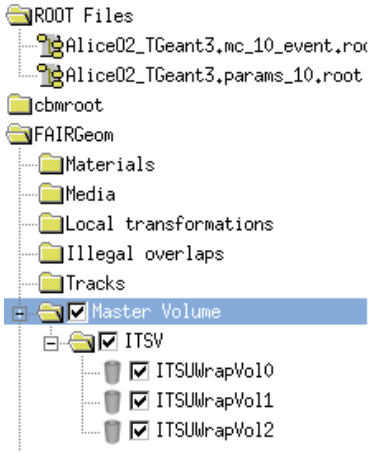


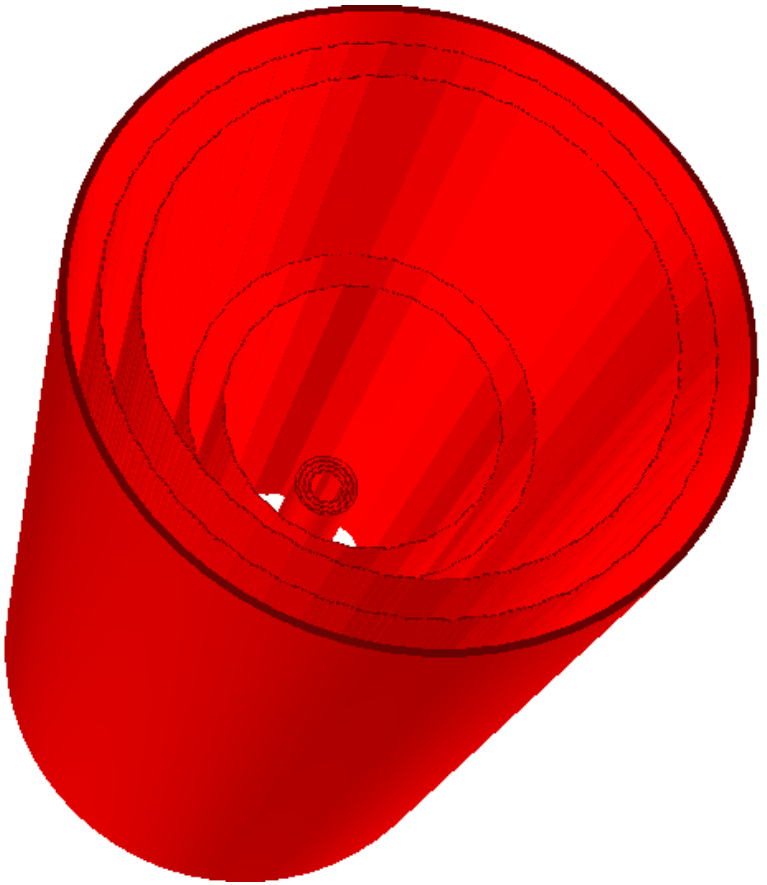
An example macro is included with AliceO2:

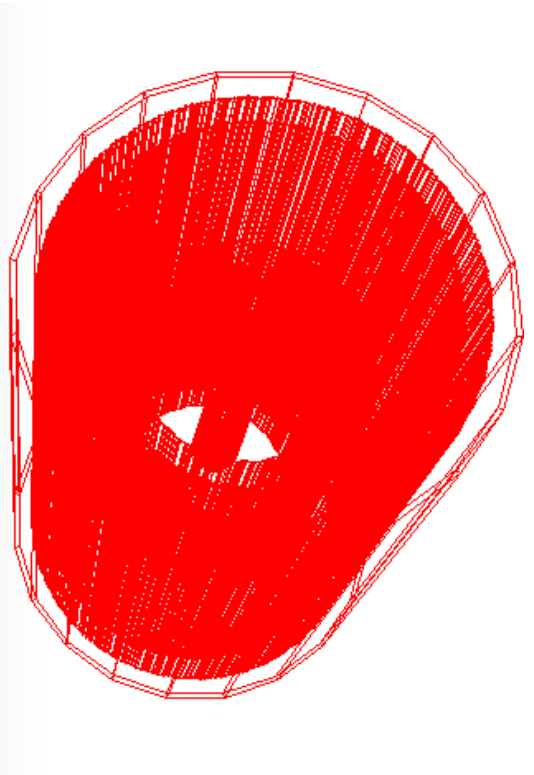
root macro/run_sim.C

The port includes the ITS Upgrade geometry

- UpgradeV1Layer (*AliITSUv1Layer*): defines the Geometry for the ITS Upgrade using Tgeo
- V11Geometry (*AliITSv11Geometry*): is a base class for the ITS geometry version 11
- UpgradeGeometryTGeo (*AliITSUGeomTGeo*): an interface class to TgeoManager. It is used in order to query the TGeo ITS geometry

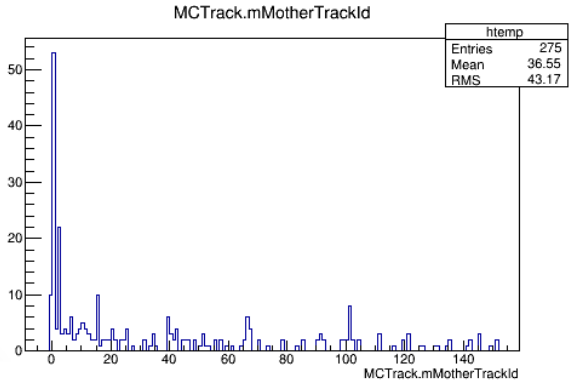


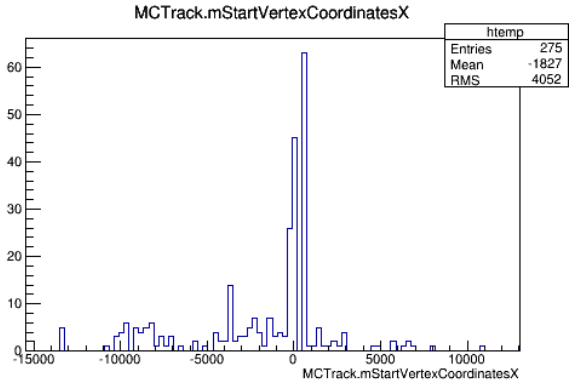


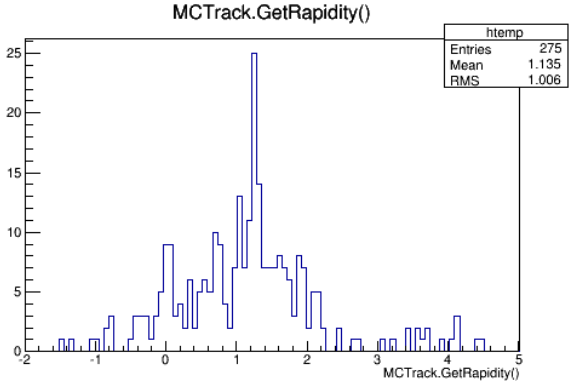


To perform simulation and create *points,* the virtual *ProcessHits* method of FairRoot is used in the Detector (*O2its*) class.

- The method is called from the MC stepping
- Information (Energy Loss, Track time, Track number Id and Volume Id) is recorded on the points
- The position and momentum of the particles is tracked from MC
- On every step of the active volume, a point is created with information on Track number Id, Volume Id, the particle entrance position, the current position, the momentum, the entrance time, the current time, the length and the energy loss
- The points are added to an AliceO2::Data::Stack stack







Testing infrastructure:

- Testing capability was recently added for the ITS simulation (Mohammad)
- A small script has to be added to CmakeLists.txt:

```
GENERATE_ROOT_TEST_SCRIPT(${CMAKE_SOURCE_DIR}/macro/run_sim.C)

ForEach(_mcEngine IN ITEMS TGeant3 TGeant4)
  Add_Test(run_sim_${_mcEngine}
          ${CMAKE_BINARY_DIR}/macro/run_sim.sh 10 \"${_mcEngine}\")
  Set_Tests_Properties(run_sim_${_mcEngine} PROPERTIES TIMEOUT "30")
  Set_Tests_Properties(run_sim_${_mcEngine} PROPERTIES PASS_REGULAR_EXPRESSION "Macro finished
succesfully")
EndForEach(_mcEngine IN ITEMS TGeant3 TGeant4)
```

- It is invoked by running *make test*

Testing infrastructure:

- It is executing the macro/run_sim.C macro using Geant3 and Geant4:

Start 1: run_sim_TGeant3
1/2 Test #1: run_sim_TGeant3 ..... Passed 12.13 sec
Start 2: run_sim_TGeant4
2/2 Test #2: run_sim_TGeant4 ..... Passed 7.15 sec

Part of an automate procedure to publish test results to *CDash*

AliceO2

Dashboard Calendar Previous Current Project

No file changed as of **Wednesday, November 19 2014 - 02:00 CET**

Nightly

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
alinsure.cern.ch	Ubuntu-linux-x86_64-gcc4.8-fairroot_-fairsoft_FairSoft_dev	0	1	0	1	1	0	0	0
pb-d-128-141-184-55	MacOS-darwin-x86_64-gcc4.2.1-fairroot_-fairsoft_FairSoft_dev	0	0	1	0	15	0	0	2

Continuous

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
alinsure.cern.ch	Ubuntu-linux-x86_64-gcc4.8-fairsoft_14.11	0	0	0	0_{-1}	2^{+2}_{-1}	0	1^{+1}	1^{+1}
alinsure.cern.ch	Ubuntu-linux-x86_64-gcc4.8-fairsoft_14.11	0	1	0	1	1	0	0	0

Experimental

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
alinsure.cern.ch	Ubuntu-linux-x86_64-gcc4.8-fairroot_-fairsoft_FairSoft_dev	0	0	0	0	1	0	0	2
macpromat	MacOS-darwin-x86_64-gcc4.2.1-fairroot_-fairsoft_FairSoft_dev	0	0	1	0_{-1}	0_{-1}	0	0	2^{+2}
macpromat	MacOS-darwin-x86_64-gcc4.2.1-fairroot_-fairsoft_FairSoft_dev	0	1	0	1^{+1}	1^{+1}_{-2}	0	0_{-2}	0
macpromat	MacOS-darwin-x86_64-gcc4.2.1-fairroot_-fairsoft_FairSoft_dev	0	0	1	0	2	0	2	0
pb-d-128-141-184-55	MacOS-darwin-x86_64-gcc4.2.1-fairroot_-fairsoft_FairSoft_dev	0	0	1	0	2	0	0	2

Testing infrastructure:

- It is executing the macro/run_sim.C macro using Geant3 and Geant4:

Start 1: run_sim_TGeant3
1/2 Test #1: run_sim_TGeant3 ..... Passed 12.13 sec
Start 2: run_sim_TGeant4
2/2 Test #2: run_sim_TGeant4 ..... Passed 7.15 sec

Part of an automate procedure to publish test results to *CDash*

```
-- Looking for GEANT4VMC... - found  /Users/turany/fairsoft/install/alfa/lib
-- Looking for VGM...
-- Looking for VGM... - found  /Users/turany/fairsoft/install/alfa/lib
-- Looking for CLHEP...
-- Looking for CLHEP... - found /Users/turany/fairsoft/install/alfa/lib
-- Looking for CERNLIB...
-- Looking for HepMC ...
-- Looking for HepMC... - found /Users/turany/fairsoft/install/alfa/lib
-- Looking for Boost ...
-- Boost version: 1.54.0
running /bin/chmod u+x  /Users/turany/fairsoft/AliceO2/build/macro/run_sim.sh  2>&1
-- Configuring done
CMake Warning (dev):
  Policy CMP0042 is not set: MACOSX_RPATH is enabled by default.  Run "cmake
  --help-policy CMP0042" for policy details.  Use the cmake_policy command to
  set the policy and suppress this warning.

  MACOSX_RPATH is not specified for the following targets:

   ALICEHLT
   AliceO2Base
   FLP2EPNex
   FLP2EPNex_distributed
   FLP2EPNex_dynamic
   Field
   O2Data
   O2Gen
   Passive
   its

This warning is for project developers.  Use -Wno-dev to suppress it.

-- Generating done
-- Build files have been written to: /Users/turany/fairsoft/AliceO2/build
```

Testing infrastructure:

To submit a specific configuration to CDash:

```
#!/bin/bash
export LINUX_FLAVOUR=MacOS
export FAIRSOFT_VERSION="FairSoft_dev"
export SIMPATH=/Users/turany/fairsoft/git/install/fairsoft_dev/
export BUILDDIR=/Users/turany/fairsoft/git/AliceO2/build
export SOURCEDIR=/Users/turany/fairsoft/git/AliceO2
export FAIRROOTPATH=/Users/turany/fairsoft/git/install/v-14.11/
export NCPU=8
```

To submit the configuration to CDash:

./Dart.sh Experimental CONFIGNAME

Questions?