

Table of Contents

Resources.....	1
SVN Repo.....	1
Doxygen.....	1
Development.....	2
Documentation.....	3
CMT Tricks.....	3
Using an external shared object and header files in a program.....	3
How to use library to read events.....	3
Reading events from a data file.....	3
Reading events online.....	4
Geometry Class for detectors.....	4
Contacts.....	6

Resources

SVN Repo

To checkout analysis code:

```
svn co svn+ssh://svn.cern.ch/repos/bl4sextras
```

One can see the websvn interface from the following address:

https://svnweb.cern.ch/cern/wsvn/bl4sextras/BL4S_analysis/trunk/

Doxygen

The source code is automatically documented at: <http://test-bl4sdoc.web.cern.ch/test-bl4sdoc/>

Development

Documentation

The analysis library is documented using doxygen here: <http://test-bl4sdoc.web.cern.ch/test-bl4sdoc/>

CMT Tricks

Using an external shared object and header files in a program

Lets say we want to compile myApp with libSomething, using some header files from "/path/to/inc". In our case libSomething is the library that contains the BL4S analysis classes. Following should be added to the requirements file

```
pattern -global default_include_path include_dirs ${<package>_root}/${<package>} "/path/to/inc/"

path_append LD_LIBRARY_PATH "/path/to/lib/"

application myapp myApplication.cc
macro myaplinkopts "-L/path/to/lib/ -lSomething"
```

How to use library to read events

Reading events from a data file

```
1 //Opening config file and setting configuration
3 BL4SConfig * cfg = new BL4SConfig("test.cfg");
5
7 //Creating instance of debug object, which handles verbosity
9 BL4SDebug * debug = new BL4SDebug(cfg->GetVerbosity());
11
13 //Opening data file
15 BL4SDataFile * dat = new BL4SDataFile(datafilename, cfg);
17
19 //Getting important parameters
21 int Nevents = dat->GetNumberOfEvents();
23 int Ndwc = cfg->GetNDwc();
25 int Nlg = cfg->GetNLeadGlass();
27 int Nscin = cfg->GetNScintillator();
29
31
33 // Creating Calibration instance (to get some calibration constants and pedestal levels of
35 BL4SCalibration * cal = new BL4SCalibration(cfg, dat);
37
39 vector<u_int> rawev;
41
43 //Main loop
45 for(int k=0; k<Nevents; k++)
47 {
49 rawev = dat->GetNextRawEvent();
51
53 //Creating BL4SEvent instance (that has information of signal on each detector)
55 unique_ptr<BL4SEvent> ev(new BL4SEvent(rawev, cfg, cal));
57
59 // Here one can do anything with pointer ev such as:
61 int nevent = ev->GetEventNumbers();
63
65 for(int i=0; i<Nlg; i++)
67 {
69 lg[i].energy = ev->GetLeadGlassEnergy(i);
71 }
73
```

```
75 }
```

Reading events online

```
1 //Opening config file and setting configuration
3 BL4SConfig * cfg = new BL4SConfig("test.cfg");
5
7 //Creating instance of debug object, which handles verbosity
9 BL4SDebug *debug = new BL4SDebug(cfg->GetVerbosity());
11
13 //Getting important parameters
15 int Ndwc = cfg->GetNDwc();
17 int Nlg = cfg->GetNLeadGlass();
19 int Nscin = cfg->GetNScintillator();
21
23 // Creating Calibration instance (to get some calibration constants and pedestal levels of
25 // Note that this time Datafile argument is not entered.
27 BL4SCalibration *cal = new BL4SCalibration(cfg);
29
31 vector<u_int> rawev;
33
35 //Main loop
37 while(1)
39 {
41 // One should copy raw data of next event from the DAQ into the rawev vector. Here I use
43 rawev = getNextRawEvent_dummy();
45
47 //Creating BL4SEvent instance (that has information of signal on each detector)
49 unique_ptr<BL4SEvent> ev(new BL4SEvent(rawev, cfg, cal));
51
53 // Here one can do anything with pointer ev such as:
55 int nevent = ev->GetEventNumbers();
57
59 for(int i=0; i<Nlg; i++)
61 {
63 lg[i].energy = ev->GetLeadGlassEnergy(i);
65 }
67
69 }
```

Geometry Class for detectors

Geometry Class is finished.

One should define a config a class first which will be input for the constructor of the Geometry class. from the config file number of detectors, verbosity and config file path is taken.

One can use the class as:

```
1 //Opening config file and setting configuration
3 BL4SConfig * cfg = new BL4SConfig("test.cfg");
5
7 //Opening geometry file
9 BL4SGeometry * geo = new BL4SGeometry("geometry_test.cfg", cfg);
11 //showing info read
13 geo->ShowGeometry();
15
17 cout << "Halo counter x opening : " << geo->HaloXOpening() << endl;
19 cout << "Halo counter y opening : " << geo->HaloYOpening() << endl;
21
23 double x, y, z;
25
27 for(int i=0; i<geo->GetNLeadGlass(); i++)
```

```
29 {  
31   similar for other detectors.  
33   geo->GetLeadGlassPosition(i,x,y,z);  
35   cout << "Lead Glass " << i << ": x=" << x << ", y=" << y << ", z=" << z << endl;  
37 }
```

Contacts

- CenkYildiz - Original Author
- SaimeSarikaya - Original Author

-- CenkYildiz - 19 Mar 2014 -- CenkYildiz - 02 Jul 2014

-- SaimeSarikaya - 14 Jul 2014

This topic: BL4S > AnalysisSoftware

Topic revision: r14 - 2015-06-29 - TimBrooks



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors. Ideas, requests, problems regarding TWiki? Send feedback