psana - Tutorials

Unknown macro: 'html'

Unknown macro: 'html'

Psana user examples

This page holds a few example code-snippets for use in psana analysis. The analysis is written in C++ and uses root for plotting the data. (The histogramming interface allows for swapping root with any other histogramming package of your choice... in principle. If you know a good one, go ahead and try and let us know so we can try it too!!)

If you didn't already, take a look at psana - User Manual and psana - Reference Manual.

Note that with C++/psana (as opposed to with python/pyana), you do need two files for your source code: MyPackage/include/MyModule.h and MyPackage /src/MyModule.cpp. Also you **must** compile any time you change your source code.

BeamLine Data: EBeam

The bare outline... I will fill in more here.
This example shows how to create 1D and 2D histograms of data from the "BldInfo(EBeam)". The histograms gets saved automatically in a root-file, filename is the same as the xtc file name, but with extension .root, and it is stored in your current directory.

PlotBld.h

```
#ifndef PSANA_EXAMPLES_PLOTBLD_H
#define PSANA_EXAMPLES_PLOTBLD_H
#include "psana/Module.h"
namespace psana_examples {
class PlotBld : public Module {
public:

  // Default constructor
  PlotBld (const std::string& name) ;

  // Destructor
  virtual ~PlotBld () ;

  /// Method which is called with event data
  virtual void beginJob(Event& evt, Env& env);
  virtual void event(Event& evt, Env& env);

protected:

private:

  Source m_ebeamSrc;
  PSHist::H1* m_ebeamHisto;
  PSHist::H1* m_chargeHisto;
  PSHist::H2* m_xPosVsAngle;
  PSHist::H2* m_yPosVsAngle;

};
} // namespace psana_examples
#endif // PSANA_EXAMPLES_PLOTBLD_H
```

PlotBld.cpp

```
#include "MyPackage/PlotBld.h"        // This includes the header file above
#include "MsgLogger/MsgLogger.h"
#include "psddl_psana/bld.ddl.h"
using namespace psana_examples;
PSANA_MODULE_FACTORY(PlotBld)
namespace psana_examples {

PlotBld::PlotBld (const std::string& name)
  : Module(name)
{
  m_ebeamSrc = configStr("eBeamSource", "BldInfo(EBeam)");
}
PlotBld::~PlotBld ()
{
}

// Method which is called once at the beginning of the job
void
PlotBld::beginJob(Event& evt, Env& env)
{
  m_ebeamHisto = env.hmgr().hist1f("ebeamHisto", "ebeamL3Energy value", Axis(1000, 13000, 16000));
  m_chargeHisto = env.hmgr().hist1f("chargeHisto", "ebeamCharge value", Axis(250, 0, 0.25));
  m_xPosVsAngle = env.hmgr().hist2f("xPosVsAngle", "x position vs. angle; position []; angle [rad]",
                                    Axis(100, -0.5,0.5), // x-axis: position
                                    Axis(100,-0.5, 0.5)); // y-axis: angle
  m_yPosVsAngle = env.hmgr().hist2f("yPosVsAngle", "y position vs. angle; position []; angle [rad]",
                                    Axis(100, -0.5,0.5),  // x-axis: position
                                    Axis(100,-0.5, 0.5)); // y-axis: position
}
// Method which is called with event data
void
PlotBld::event(Event& evt, Env& env)
{
  // Assume this experiment has V1 version of EBeam data
  // how do you know? Try parsing xtc file with 'pyxtcreader |less'
  shared_ptr<Psana::Bld::BldDataEBeamV1> ebeam = evt.get(m_ebeamSrc);
  if (ebeam.get()) {
    m_ebeamHisto->fill( ebeam->ebeamL3Energy() );
    m_chargeHisto->fill( ebeam->ebeamCharge() );
    m_xPosVsAngle->fill( ebeam->ebeamLTUPosX(),ebeam->ebeamLTUAngX() );
    m_xPosVsAngle->fill( ebeam->ebeamLTUPosY(),ebeam->ebeamLTUAngY() );
//    WithMsgLog(name(), info, str) {
//      str << "Bld::BldDataEBeamV1:"
//          << "\n  damageMask=" << ebeam->damageMask()
//          << "\n  ebeamCharge=" << ebeam->ebeamCharge()
//          << "\n  ebeamL3Energy=" << ebeam->ebeamL3Energy()
//          << "\n  ebeamLTUPosX=" << ebeam->ebeamLTUPosX()
//          << "\n  ebeamLTUPosY=" << ebeam->ebeamLTUPosY()
//          << "\n  ebeamLTUAngX=" << ebeam->ebeamLTUAngX()
//          << "\n  ebeamLTUAngY=" << ebeam->ebeamLTUAngY()
//          << "\n  ebeamPkCurrBC2=" << ebeam->ebeamPkCurrBC2();
    }
  }
}
```

Displaying histograms.

To display the histograms graphically, open this root histogram file in a root session. Lauch root by typing 'root'. You can optionally load the data file by adding it on the command line. The scripting syntax is like C++,
with some additional functionality:

```
[ofte@psana0106 ana-oct]$ root e86-r0212-s00-c00.root
  ******************************************
  *                                        *
  *        W E L C O M E  to  R O O T      *
  *                                        *
  *   Version   5.24/00      29 June 2009   *
  *                                        *
  *  You are welcome to visit our Web site  *
  *          http://root.cern.ch           *
  *                                        *
  ******************************************

ROOT 5.24/00 (trunk@29257, Jun 30 2009, 09:23:51 on linuxx8664gcc)

CINT/ROOT C/C++ Interpreter version 5.17.00, Dec 21, 2008
Type ? for help. Commands must be C++ statements.
Enclose multiple statements between { }.
...using style 'Plain'
...using style 'Plain'
root [0]
Attaching file e86-r0212-s00-c00.root as _file0...
root [1] _file0.ls()       // list the contents of the file
TFile**         e86-r0212-s00-c00.root  Created by the RootHManager
 TFile*         e86-r0212-s00-c00.root  Created by the RootHManager
  KEY: TH1F     ebeamHisto;1    ebeamL3Energy value
  KEY: TH1F     chargeHisto;1   ebeamCharge value
  KEY: TH2F     xPosVsAngle;1   x position vs. angle
  KEY: TH2F     yPosVsAngle;1   y position vs. angle
root [2] ebeamHisto->Draw()
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [3] TCanvas c2   // to make another plot in a new window, create a new canvas.
root [4] xPosVsAngle->Draw() // this will be plotted onto the current canvas (last created)
root [5] chargeHist->Draw()
Error: Symbol chargeHist is not defined in current scope  (tmpfile):1:
Error: Failed to evaluate chargeHist->Draw()
*** Interpreter error recovered ***
root [6] chargeHisto->Draw()
root [7] .q   // this is how you quit root
```

? Unknown Attachment