Analysis 2
Histograms and ntuples

Paul Guèye

Hampton University

Hampton VA

gueye@jlab.org

Outline

I. Overview

a) **Downloading the AIDA utility package**

b) **Turning on the analysis package**

c) **Loading the package**

d) **Producing/Analyzing first histograms**

e) **The packages: JAS, ROOT, PAW, OpenScientist**

II. Few examples of advanced analysis

a) **JAS**

b) **ROOT**

c) **PAW**

d) **OpenScientist**

III. *“Fighting* Geant4” with Momo

IV. Conclusion

Overview

Downloading & Setting the AIDA utility package

(thanks to Guy Barrand)

- If you do have AIDA installed, don’t worry about this slide
- If you DO NOT have AIDA installed, then (**<u>will take ~5-10 min</u>**):

1. Create AIDA directory under "geant4/work" directory

   ```
   mkdir geant4/work/AIDA
   ```

2. Download "**osc_batch-v16r0-XX-i386-gcc_401.zip**" into "AIDA" folder from http://geant4.in2p3.fr/2007/AIDA/AIDA.zip

– Windows	XX = CYGWIN
– Linux	XX = Linux
– MACOS	XX = Darwin

3. Decompress the file using "**unzip command**” under unix or the **windows unzip utility**
4. Go into “osc_batch/v16r0” directory

   ```
   cd osc_batch/v16r0
   ```

5. Setup the AIDA environment

   source aida-setup.csh or **source aida-setup.sh**

6. Done!

Turning on the Analysis Package

- G4ANALYSIS_USE
- Set variable
  - setenv G4ANALYSIS_USE 1 (csh)
  - export G4ANALYSIS_USE=1 (bash)
- Is it on? (unix commands)
  - 1 (on) or 0 (off)
  - **“echo”** echo $G4ANALYSIS_USE
  - **“env”** env then look for G4ANALYSIS_USE

Loading the analysis package I

Required files

#include <AIDA/AIDA.h>	→	Define
AnalysisManager.cc	→	Create & Fill
AnalysisManager.hh		

AIDA.h (Define)

```
#include "AIDA/IAnalysisFactory.h"
#include "AIDA/IAnnotation.h"
#include "AIDA/IAxis.h"
#include "AIDA/IAxisStyle.h"
#include "AIDA/IBaseHistogram.h"
#include "AIDA/IBaseStyle.h"
#include "AIDA/IBrushStyle.h"
#include "AIDA/ICloud.h"
#include "AIDA/ICloud1D.h"
#include "AIDA/ICloud2D.h"
#include "AIDA/ICloud3D.h"
#include "AIDA/IConstants.h"
#include "AIDA/IDataPoint.h"
#include "AIDA/IDataPointSet.h"
#include "AIDA/IDataPointSetFactory.h"
#include "AIDA/IDataStyle.h"
#include "AIDA/IEvaluator.h"
#include "AIDA/IFillStyle.h"
#include "AIDA/IFilter.h"
#include "AIDA/IFitData.h"
#include "AIDA/IFitFactory.h"
#include "AIDA/IFitParameterSettings.h"
#include "AIDA/IFitResult.h"
#include "AIDA/IFitter.h"
#include "AIDA/IFunction.h"
#include "AIDA/IFunctionCatalog.h"
#include "AIDA/IFunctionFactory.h"
```

```
#include "AIDA/IHistogram.h"
#include "AIDA/IHistogram1D.h"
#include "AIDA/IHistogram2D.h"
#include "AIDA/IHistogram3D.h“
#include "AIDA/IHistogramFactory.h“
#include "AIDA/IAnalysisFactory.h"
#include "AIDA/IHistogramFactory.h"
#include "AIDA/IInfo.h"
#include "AIDA/IInfoStyle.h"
#include "AIDA/ILineStyle.h"
#include "AIDA/IManagedObject.h"
#include "AIDA/IMarkerStyle.h"
#include "AIDA/IMeasurement.h"
#include "AIDA/IModelFunction.h"
#include "AIDA/IPlotter.h"
#include "AIDA/IPlotterFactory.h"
#include "AIDA/IPlotterLayout.h"
#include "AIDA/IPlotterRegion.h"
#include "AIDA/IPlotterStyle.h"
#include "AIDA/IProfile.h"
#include "AIDA/IProfile1D.h"
#include "AIDA/IProfile2D.h"
#include "AIDA/IRangeSet.h"
#include "AIDA/ITextStyle.h"
#include "AIDA/ITitleStyle.h"
#include "AIDA/ITree.h"
#include "AIDA/ITreeFactory.h"
#include "AIDA/ITuple.h"
#include "AIDA/ITupleFactory.h"
```

Loading the analysis package II: AnalysisManager.cc (create)

From Extended Example A01

```
…
A01AnalysisManager::A01AnalysisManager()
:analysisFactory(0), hFactory(0), tFactory(0)
{
  // Hooking an AIDA compliant analysis system.
  analysisFactory = AIDA_createAnalysisFactory();
  if(analysisFactory)
  {
    ITreeFactory* treeFactory = analysisFactory->createTreeFactory();
    tree = treeFactory->create("A01.aida","xml",false,true,"compress=yes");
    hFactory = analysisFactory->createHistogramFactory(*tree);
    tFactory = analysisFactory->createTupleFactory(*tree);
    delete treeFactory; // Will not delete the ITree.
  }
}
…
```

Loading the analysis package II: AnalysisManager.cc (create)

From Extended Example AnaEx01: create histograms

```
…
 std::string opts = "compress=no";
 fTree = treeFactory->create("AnaEx01.aida","xml",false,true,opts);
…
 if(!fTree) return;
 fTree->mkdir("histograms");
 fTree->cd("histograms");

// Create an histo factory that will create histo in the tree :
  AIDA::IHistogramFactory* histoFactory = fAIDA->createHistogramFactory(*fTree);
  if (histoFactory)
{
    fEAbs = histoFactory->createHistogram1D("EAbs",100,0,100);
    fLAbs = histoFactory->createHistogram1D("LAbs",100,0,100);
    …
  delete histoFactory;
}
…
```

Loading the analysis package III: AnalysisManager.cc (create)

From Extended Example AnaEx01: create ntuples

```
…
  fTree->cd("..");
  fTree->mkdir("tuples");
  fTree->cd("tuples");

  // Get a tuple factory :
  AIDA::ITupleFactory* tupleFactory = fAIDA->createTupleFactory(*fTree);
  if (tupleFactory)
      {
      // Create a tuple :
      fTuple = tupleFactory->create("AnaEx01","AnaEx01",
      "double EAbs,double LAbs,double EGap,double LGap");

      delete tupleFactory;
  }
…
```

Loading the analysis package IV: AnalysisManager.cc (fill)

From Extened Example AnaEx01: fill histograms and ntuples

```
…
 if (CHC)
{
    G4int n_hit = CHC->entries();
    for (G4int i=0;i<n_hit;i++) {
      G4double EAbs = (*CHC)[i]->GetEdepAbs();
      G4double LAbs = (*CHC)[i]->GetTrakAbs();
      …
      fEAbs->fill(EAbs);
      fLAbs->fill(LAbs);
      …

      fTuple->fill(0,EAbs);
      fTuple->fill(1,LAbs);
      …
      fTuple->addRow();
    }
  }…
```

Producing/Analyzing first histograms

From **example A01**

1. Copy example into your work area:
   cp –r $G4INSTALL/examples/extended/A01app ~/geant4/work/.
2. Go into example directory: **cd A01app**
3. Compile: **gmake** or **make**
4. Run: **~/geant4/work/bin/Linux-g++/A01app aida.mac**
5. A new file should appear: **A01.aida**
6. Start jas
7. Open the file: **File → "Open File"** then **select A01.aida**
8. Double click on "**Blue Folder**" (A01.aida)
9. Double click on histogram of interest

Few examples of advanced analysis

- Java Analysis Studio: JAS (all files)
- Physics Analysis Workstation: PAW (hbook files)
- The Root of Everything: ROOT (root files)
- OpenScientist (all files)
- From **example AnaEx01**
  - Copy example into your work area:

    cp –r $G4INSTALL/examples/extended/AnaEx01 ~/geant4/work/.
  - Go into example directory: **cd AnaEx01**
  - Compile: **gmake** or **make**
  - Edit **analysis/jas/run.mac** file: **/run/BeamOn 100**
  - Run: **~/geant4/work/bin/Linux-g++/AnaEx01**

Java Analysis Studio (JAS, http://jas.freehep.org/jas3)

- In AnaEx01AnalysisManager.cc
  → **Verify or type the following lines**
  std::string opts = "compress=no";
  fTree = treeFactory->create("AnaEx01.aida","xml",false,true,opts);

- cd analysis/jas
- ~/geant4/work/bin/Linux-g++/AnaEx01 run.mac
- Should get: **AnaEx01.aida**
- Start jas
- Open the file: **File → “Open File”** then **select AnaEx01.aida**
- Double click on “**Blue Folder**” (AnaEx01.aida)
- Double click on **histogram** and then the **ones of interest**
- Double click on **tuples** then “Yellow Folder” (AnaEx01)

ROOT (http://root.cern.ch/)

- In AnaEx01AnalysisManager.cc
  → **Verify or type the following lines**
  ```
  std::string opts = “export=root";
  fTree = treeFactory->create("AnaEx01.root",“ROOT",false,true,opts);
  ```

- cd analysis/jas
- ~/geant4/work/bin/Linux-g++/AnaEx01 run.mac
- Should get: **AnaEx01.root**
- Start root
- Launch browser: new TBrowser()
- Open the file: **File → “Open”** then **select AnaEx01.root**
- Double click on **“Yellow Folder”** (ROOT Files)
- Double click on **“Yellow Folder”** (AnaEx01.root)
- Double click on **“Yellow Folder”** (histograms) and then **the ones of interest**
- Double click on **“Yellow Folder”** (tuples) then **right-mouse** on AnaEx01;1 and select **Start Viewer**
- Drag variables of interest into (X,Y,Z) (Ex: **X=EAbs and Y=Labs**)
- Select **Draw Icon** (bottom left corner)
- Create a cut: double click on **E()** then type: **EGapcut** (alias) and **EGap<20** (expression)
- Drag cut into **Scissor**
- Select **Draw Icon** (bottom left corner)
- Can **enable** or **disable** the **cut** by **double clicking** on it

Physics Analysis Workstation (PAW, cernlib)

PAW is no longer supported: you are on your own!!!

- In AnaEx01AnalysisManager.cc
  → **Verify or type the following lines**
  std::string opts = “export=hbook";
  fTree = treeFactory->create("AnaEx01.hbook",“HBOOK",false,true,opts);

- cd analysis/jas
- ~/geant4/work/bin/Linux-g++/AnaEx01 run.mac
- Should get: **AnaEx01.hbook**
- Start paw++
- Open file: **h/file 0 [filepath]/AnaEx01.hbook**
- Double click on **logical unit 1** (lun1)
- Double click on **histograms** or **ntuple**
- Select a variable and double lick in (X,Y,Z) (Ex: **X=EAbs** and **Y=Labs**)
- Click on **Plot**
- Click on **Cut Editor …**
- Left box = variable, middle box = expression, right box = value (Ex: EGap<20)
- Click on **Plot**

OpenScientist (http://openscientist.lal.in2p3.fr/)

- Start OpenScientist via **onx** command
- Open **file of interest** under **File menu**
- Self explanatory from this point

Fighting Geant4 with MOMO (http://erpc1.naruto-u.ac.jp/~geant4/)

Famous Japanese Samurai who used to fight Giants!!

- Needs GGE, GAG & GPE jar files
- Naruto.jp/momo
- Everything is a drag & drop!!
- Cannot do complicated geometries
- Does not have most features of current Geant4
- An example from JLab

Conclusion

- Several analysis tools
- Geant4 does not provide the *“standard”*
- Use as it suits your needs
- Best sources
  - **users hypernews**
  - **analysis tools sites**