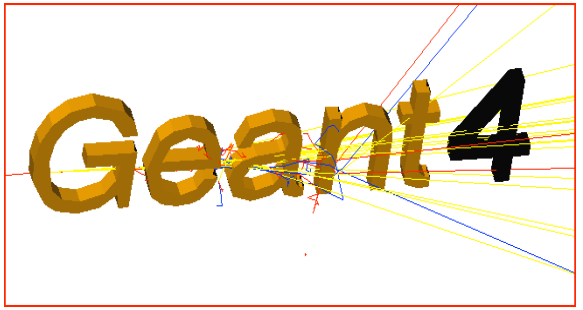



IPN
INSTITUT DE PHYSIQUE NUCLÉAIRE
ORSAY

# Analysis

I. Hrivnacova, IPN Orsay

Geant4 tutorial for ED MIPEGE,
13 - 24 May 2013, Orsay

# Acknowledgment


- Some of the slides shown were originally created for previous Geant4 tutorials and provided with courtesy for this course
- We wish to thank our Geant4 colleagues for allowing us to re-use their material
- The credits for slides re-used in this presentation
  - Joseph Perl & Michael Kelsey, SLAC, G4 tutorial at KISTI, 2010


SLAC NATIONAL ACCELERATOR LABORATORY

# Outline

- History
- Geant4 analysis tools
  - Histogram, ntuples, analysis commands
- Plotting With External analysis tools
  - ROOT, Gnuplot, Excel, Open[Libre]Office, OpenScientist, ...
- Interfacing With External Analysis Tools
  - ROOT, AIDA

# History

- Historically first analysis tools in Geant4 examples
- Based on AIDA = Abstract Interfaces for Data Analysis
  - First provided within the Geant4 example extended/analysis/AnaEx01 (jas, Lab), then available as external tools
  - The AIDA compliant tools (linked in the Geant4 Guide for Application Developers):
    - JAS, iAIDA, Open Scientist Lab, rAIDA
  - Not all kept maintained, not all implement the AIDA interfaces completely
  - Not always easy to be installed & used
    - See Geant4 user forum, Analysis category
  - Still supported with Geant4 9.6 (November 2012)

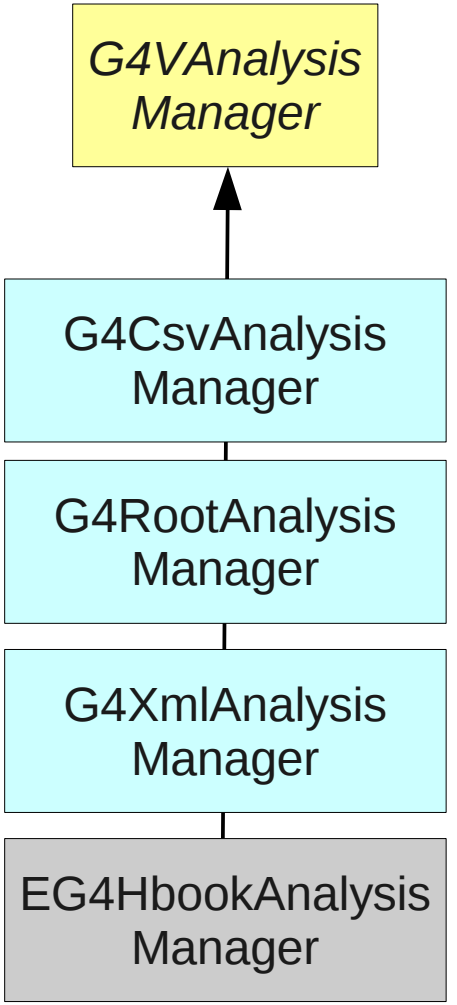
# New Analysis Tools

- New analysis category in Geant4 since Geant4 9.5 (December 2011)
- Based on g4tools from inlib/exlib developed by G. Barrand (LAL):
  - http://inexlib.lal.in2p3.fr/
  - “Pure header code” - all code is inlined
  - Can be installed on iOS, Android, UNIXes, Windows
  - Provides code to write histograms and “flat ntuples” in several formats: ROOT, XML AIDA format, CSV for ntuples. HBOOK
- Complete migration to g4tools in all Geant4 examples is in the development plan for 2013

# Analysis Managers

Provide

- Uniform interface to g4tools
  - Hide the differences according to a selected technology (root, XML, HBOOK) from the user
- Higher level management of g4tools objects (file, histograms, ntuples)
  - Memory management
  - Access to histograms, ntuple columns via indexes
- Integration in the Geant4 framework
  - Interactive commands, Units



# Histograms

B4RunAction.cc

```
#include "B4Analysis.hh"

void B4RunAction::BeginOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* man = G4AnalysisManager::Instance();

  // Open an output file
  man->OpenFile("exampleB4");

  // Create histogram(s)
  man->CreateH1("0","Edep in absorber", 100, 0., 800*MeV);
  man->CreateH1("1","Edep in gap", 100, 0., 100*MeV);
}

void B4RunAction::EndOfRunAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->Write();
  man->CloseFile();
}
```

B4Analysis.hh

```
#ifndef B4Analysis_h
#define B4Analysis_h 1

#include "g4root.hh"
//#include "g4xml.hh"
//#include "g4hbook.hh"

#endif
```

*Selection of the output format at a single place*

B4EventAction.cc

```
#include "B4Analysis.hh"

void N4EventAction::EndOfEventAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->FillH1(0, fEnergyAbs);
  man->FillH1(1, fEnergyGap);
}
```

*Histogram IDs are attributed Automatically*

# More On Histograms

- Histogram Identifiers
  - The histogram ID is automatically generated when a histogram is created by G4AnalysisManager::CreateH1(), and its value is returned from this function.
    - The histogram names ("0", "1") have no relation to the histogram ID which is used afterwords in histograms fill.
  - The default start value 0 can be changed (eg. to 1) with the G4AnalysisManager->SetFirstHistoId(G4int) method.
  - The 1D and 2D histigrams IDs are defined independently
- Histogram Objects
  - It is also possible to access directly the histogram by G4AnalysisManager::GetH1(G4int id). The concrete histogram type is hidden behind a selected namespace

```
G4cout << "\n ----> print histograms statistic \n" << G4endl;
G4cout << " EAbs : mean = " << analysisManager->GetH1(1)->mean()
       << " rms = " << analysisManager->GetH1(1)->rms() << G4endl;
```

# Activation Of Histograms

- The activation option allows the user to activate only selected histograms.
- When this option is activated, only the histograms marked as activated are returned, filled or saved in a file.
  - Note that by default this feature is  not activated
  - This feature is intensively used in extended/electromagnetic examples where all histograms are first created inactivated

```
G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();
 analysisManager->SetActivation(true);
  // define histogram parameters name, title, nbins, vmin, vmax
  G4int id = analysisManager->CreateH1(name, title, nbins, vmin, vmax);
  analysisManager->SetActivation(G4VAnalysisManager::kH1, id, false);
```

- Then only selected histograms are activated in macros, using the analysis "set" command

# Histograms Properties

- The properties, additional to those defined in g4tools, can be added to histograms via G4AnalysisManager
- *Unit*: if a histogram is defined with a unit, all filled values are automatically converted to this defined unit and the unit is added to the histogram axis title.
- *Function*: if a histogram is defined with a function, the function is automatically executed on the filled values and its name is added to the histogram axis title.
  - The available functions: log, log10, exp .
  - When a histogram is defined with both unit and function the unit is applied first.
- *Activation*: see previous slide.
- *ASCII option*: if activated the histogram is also printed in an ASCII file when G4AnalysisManager::Write() function is called.

# Analysis Commands

- A set of Geant4 commands which can be used to create histograms or set their properties dynamically
  - They are implemented in G4AnalysisMessenger class
- Commands for handling files, directories and general options

```
/analysis/setFileName name        # Set name for the histograms and ntuple file
/analysis/setHistoDirName name    # Set name for the histograms directory
/analysis/setNtupleDirName name   # Set name for the histograms directory
/analysis/setActivation true|false # Set activation option
/analysis/verbose level           # Set verbose level
```

# Analysis Commands (2)

- Commands to create or define 1D histogram:

```
/analysis/h1/create
   name title [nbin min max] [unit] [fcn]          # Create 1D histogram
/analysis/h1/set id nbin min max [unit] [fcn]    # Set parameters
```

- Commands to create or define 2D histogram:

```
/analysis/h2/create                                          # Create 2D histogram
   name title [nxbin xmin xmax xunit xfcn nybin ymin ymax yunit yfcn]
/analysis/h2/set                                             # Set parameters
   id nbin xmin xmax xunit xfcn nybin ymin ymax yunit yfcn
```

- Example of a macro gammaSpectrum.mac  in TestEm5

```
/analysis/setFileName gammaSpectrum
/analysis/h1/set  3 200 0.01 10 MeV          #gamma: energy at vertex
/analysis/h1/set  5 200 0.01 10 MeV log10  #gamma: energy at vertex (log10)
/analysis/h1/set 20 200 0 6 MeV               #gamma: energy at exit
/analysis/h1/set 40 200 0 6 MeV               #gamma: energy at back
```

# Analysis Commands (3)

- The commands for 1D histogram control:

```
/analysis/h1/setAscii id true|false      # Print 1D histogram on ASCII file
/analysis/h1/setTitle id title           # Set title for the 1D histogram
/analysis/h1/setXaxis id title           # Set x-axis title for the 1D histogram
/analysis/h1/setYaxis id title           # Set y-axis title for the 1D histogram
/analysis/h1/setActivation id true|false      # Set activation for the 1D histogram
/analysis/h1/setActivationToAll  true|false   # Set activation to all 1D histograms
```

- The same set of commands is available for 2D histograms, under "/analysis/h2" directory.

# Ntuples

B4RunAction.cc

```
#include "B4Analysis.hh"

void B4RunAction::BeginOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* man = G4AnalysisManager::Instance();

  // Open an output file
  man->OpenFile("exampleB4");

  // Create ntuple
  man->CreateNtuple("B4", "Edep and TrackL");
  man->CreateNtupleDColumn("Eabs");
  man->CreateNtupleDColumn("Egap");
  man->FinishNtuple();
}
```

B4EventAction.cc

```
#include "B4Analysis.hh"

void B4EventAction::EndOfEventAction(const G4Run* aRun)
{
  G4AnalysisManager* man = G4AnalysisManager::Instance();
  man->FillNtupleDColumn(0, fEnergyAbs);
  man->FillNtupleDColumn(1, fEnergyGap);
  man->AddNtupleRow();
}
```

# More On Ntuples

- Ntuple and Ntuple Column Identifiers
  - In a similar way as for histogram ID, the ntuple and ntuple colum IDs are automatically generated when the ntuple or ntuple column is created by G4AnalysisManager::CreateNtuple() or G4AnalysisManager::CreateNtupleTColumn() and its value is returned from this function.
  - The default start value 0 can be again changed with the G4AnalysisManager::SetFirstNtupleId(G4int) method.
  - The ntuple column ID is not specific to the ntuple column type and its value is incremented in the order as the ntuple columns are created.

# Viewing/Processing Resulted Files

- The analysis tool allow to fill histograms and/or ntuples and save them in files of supported formats:
  - ROOT, XML, CSV, HBOOK
- Users Geant4 application need not to be linked with the external analysis tools in order to use the Geant4 analysis tool and produce the file(s) with histograms and/or ntuples
- But the analysis tools have to be installed on the users machines in order to view or process the analysis of the data in the generated files

# File Formats

- ROOT:
  - Can be view/processed with ROOT: http://root.cern.ch; which is able to process also the HBOOK and CSV formats
- XML (AIDA)
  - JAS, iAIDA, Open Scientist Lab, rAIDA – see more details in the Appendix 2, in Geant4 Application Developer's Guide
  - And “new” inlib/exlib: http://inexlib.lal.in2p3.fr/
- CSV (comma-separated values)
  - The simplest possible output format, can be analyzed by many tools
  - Gnuplot, Excel, OpenOffice, ROOT

# File Formats (2)

- HBOOK
  - This format was defined in a Fortran callable package (HBOOK) for histogramming and fitting which was developed in 1970~1990:
  - PAW (Physics Analysis Workstation) - provides interactive graphical presentation and statistical or mathematical analysis of the HBOOK histograms/ntuples
  - Based on several components of the CERN Program Library
  - Now fully superseded with ROOT

# Analysis of Generated Files With External Tools

# Plotting ROOT files ... with ROOT

# ROOT

ROOT is a powerful analysis tools which provides

- histogramming and graphing to view and analyze distributions and functions
- curve fitting (regression analysis) and minimization of functionals, statistics tools used for data analysis,
- matrix algebra, four-vector computations, standard mathematical functions, multivariate data analysis, e.g. using neural networks,
- persistence and serialization of objects, which can cope with changes in class definitions of persistent data, creating files in various graphics formats, like PostScript, PNG, SVG
- 3D visualizations (geometry), image manipulation, used, for instance, to analyze astronomical pictures
- access to distributed data (in the context of the Grid), distributed computing, to parallelize data analyses, access to databases,
- interfacing Python and Ruby code in both directions, interfacing Monte Carlo event generators

# Viewing ROOT Files

- Start ROOT session

```
$> root
```

- Open a ROOT browser in the ROOT interactive shell

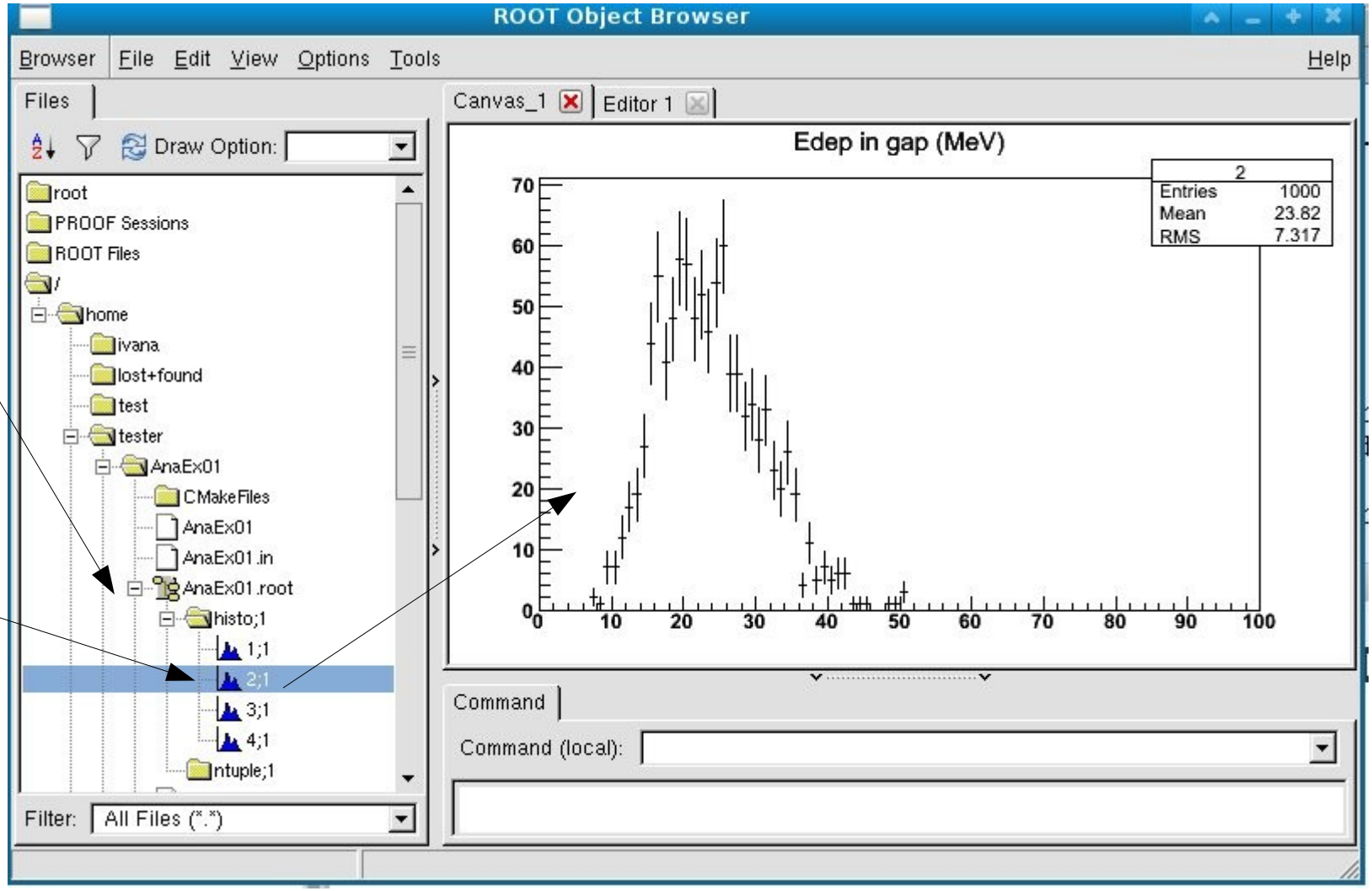
```
root [0] TBrowser b;
```

- See ROOT documentation
  - How to edit histogram properties
  - How to open Fit panel
  - How to write ROOT macros

# Viewing ROOT Files (2)

File generated
In Geant4
simulation

Selected H1
is automatically
drawn in the
canvas



# Analysis of Generated Files
# With External Tools

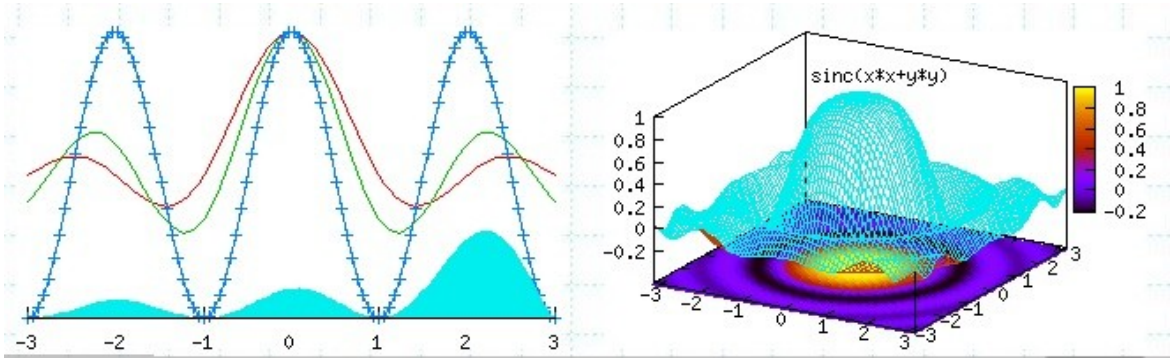
# Plotting CSV Files
# GNUplot, Excel, Open[Libre]Office

# Plotting Csv with Gnuplot

- *"Gnuplot is a portable command-line driven interactive data and function plotting utility for UNIX, IBM OS/2, MS Windows, DOS, Macintosh, VMS, Atari and many other platforms."*
- *"The software is copyrighted but freely distributed (i.e., you don't have to pay for it)."*
- http://www.gnuplot.info/
- At UNIX command line

  $> gnuplot

  plot mydata.dat




Note that files extension is .dat (and not .csv as generated from simulation)

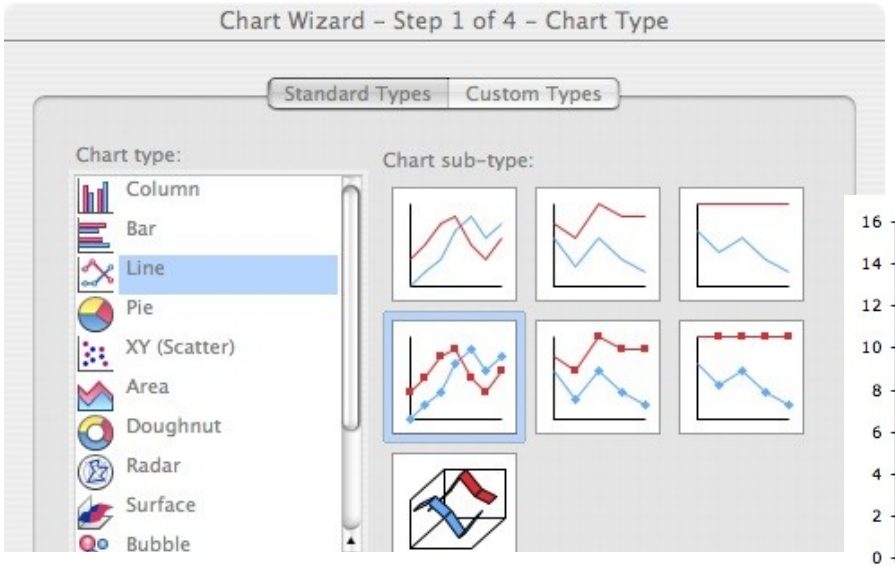
- Add info about External tools:
- ROOT, Tbrowser, Macro
- Csv – GNUplot, Excel, OpenOffice (Let's learn it !)
- XML – just links to the tools

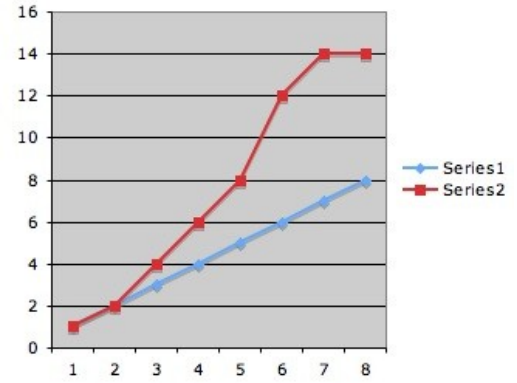
# Plotting with Excel - 1

- Here are the instrctuons to import a .csv file into Excel
- From the “Data” menu, select “External Data”…“Import a text file…”
  - Select the file
  - The Text Import Wizard will come up
  - Check the “Delimited” box
  - Hit “Next”
  - Check the “Comma” box
  - The bottom part of the Wizard should now show your data nicely separated into appropriate columns
  - Hit “Finish”
  - You will see a dialog titled “Import Data” which will ask where to put the data.  Accept the defaults:
  - Existing worksheet
  - =$A$1
  - Hit “OK”
- The data should now show up in the spreadsheet

# Plotting with Excel - 2

- Now that the data is in the spreadsheet, you can plot it as follows
  - Select at least one item in the filled data area
  - From the “Insert” menu, select “Chart…”
  - Select Chart type: “Line”
  - Hit “Finish”
  - You’re done



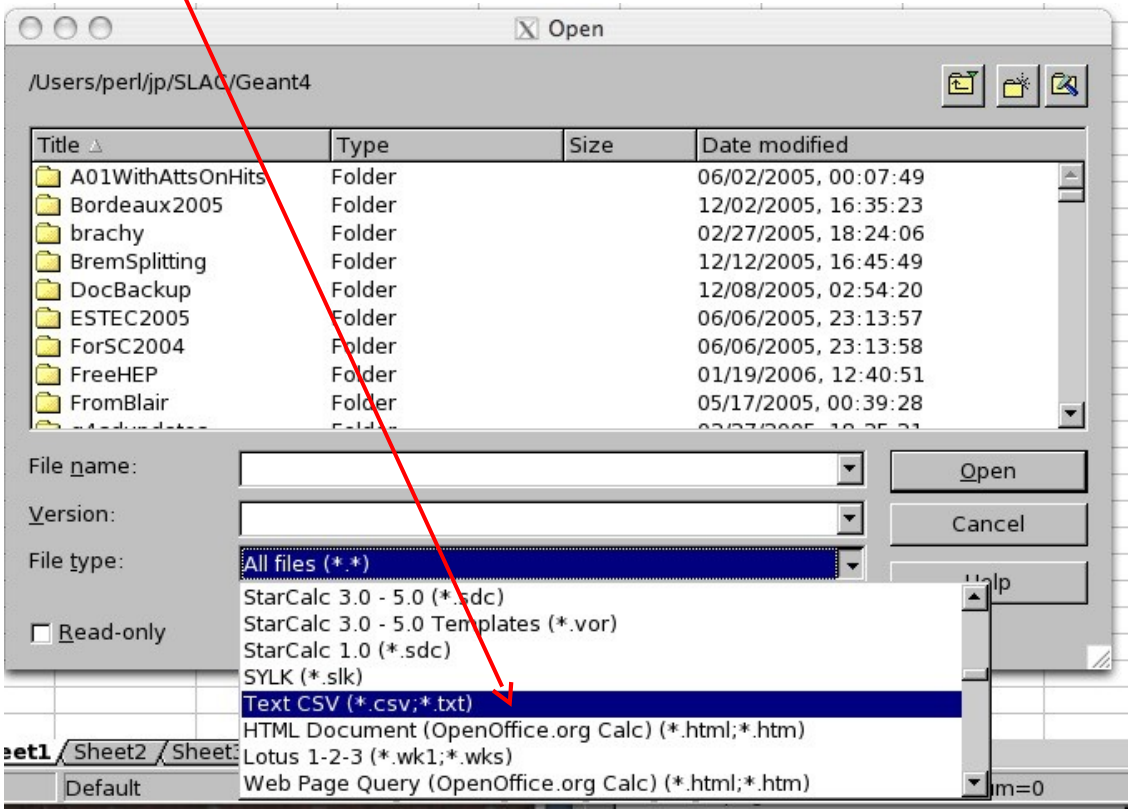



# Plotting with Open[Libre]Office - 1

- OpenOffice.org is a multiplatform and multilingual office suite and an open-source project
- LibreOffice is descended from OpenOffice, from which it was forked in 2010.
- Compatible with all other major office suites, the product is free to download, use, and distribute
- http://www.openoffice.org/
  http://www.libreoffice.org

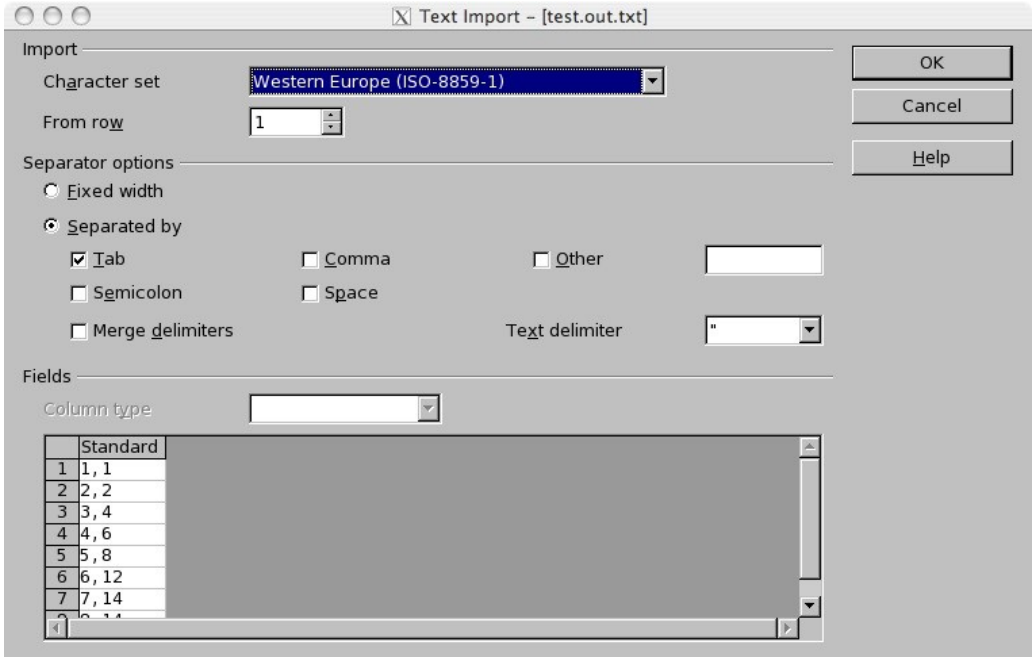
# Plotting with Open[Libre]Office - 2

- From the “File” menu, select “Open…”
  - In the “File type” box, select “Text CSV”
    Note that this is NOT the same as just “Text”
  - Then select your file

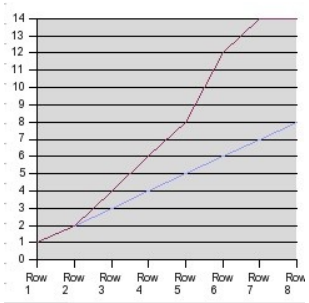


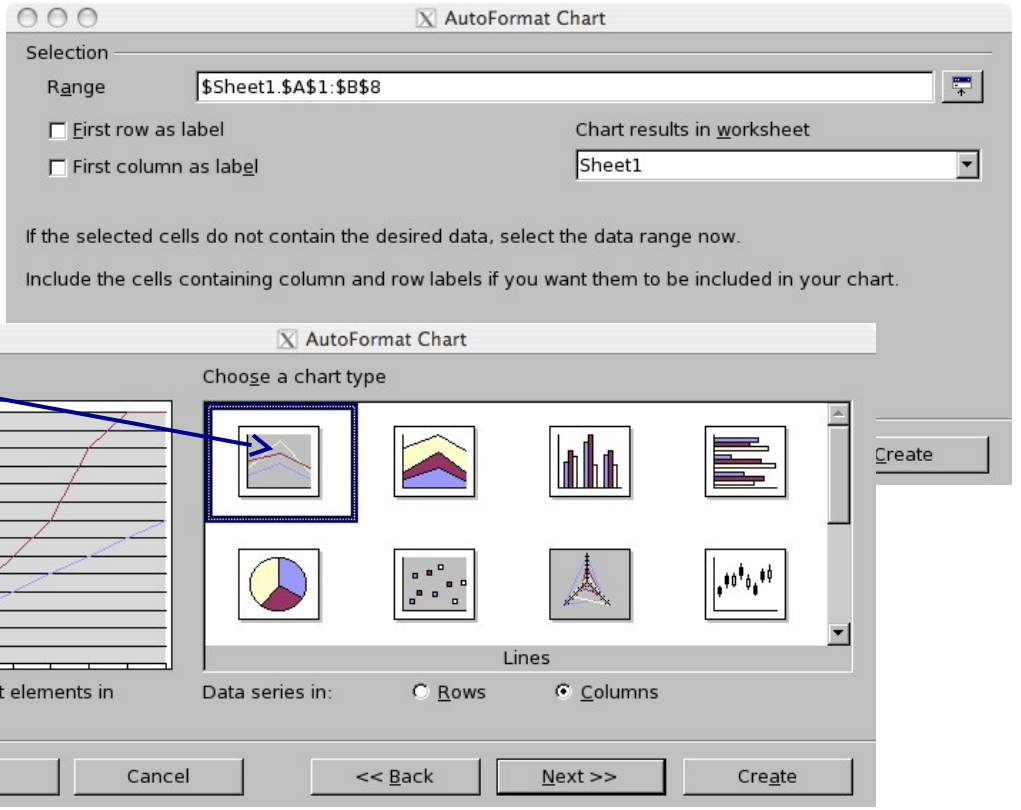
# Plotting with Open[Libre]Office - 3

- Once you select your file
  - In the Text Import dialog, which will come up,  check the “Comma” box
  - The bottom part of the dialog should now show your data nicely separated into appropriate columns.
  - Hit “OK”
- The data should now show up in the spreadsheet.



# Plotting with Open[Libre]Office - 4

- Now that the data is in the spreadsheet, you can plot it as follows
  - Select at least one item in the filled data area
  - From the “Insert” menu, select “Chart…”
  - The defaults should be fine
  - Hit “Next”
  - Select the icon for “Line” chart
  - Hit “Create”
  - 



# Interfacing With External Analysis Tools

# ROOT

# Examples With ROOT

- The Geant4 applications with use of ROOT classes are demonstrated in extended examples:
  - analysis/AnaEx02 – demonstration of use of Root histograms and ntuples
    - AnaEx01 – same with g4tools; AnaEx03 – same with AIDA
    - http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples_doc/html/Examples_analysis.html (link)
  - persistency/P01, P02
    - Root I/O examples for storing and retrieving calorimeter hits (P01) and geometry objects (P02)
    - Storing objects in a file using the 'reflection' technique for persistency provided by the Reflex tool
    - *The generation of the Reflex dictionary fails for Geant4 geometry classes using c-array with dynamic size declared via a variable of size_t type (as Reflex requires int) and therefore saving the Geant4 geometry with ROOT I/O is currently not possible*
    - Reviewing these examples with changes for Geant4 MT

# Interfacing With External Analysis Tools

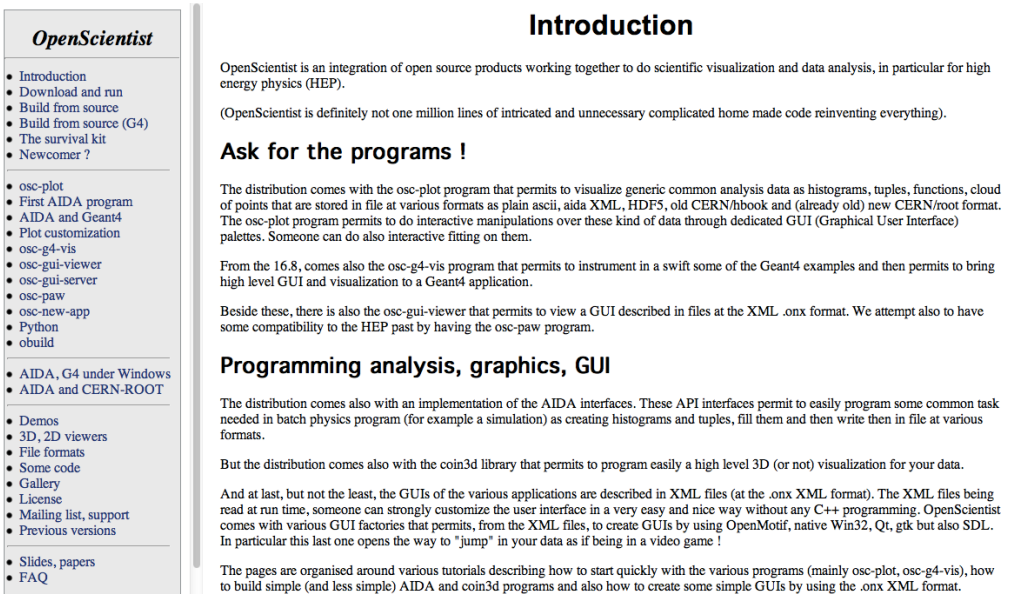
# AIDA Based Tools

# Examples With AIDA

- The Geant4 applications with use of AIDA based analysis is demonstrated in the extended example:
  - analysis/AnaEx03 – demonstration of use of AIDA histograms and ntuples
    - AnaEx01 – same with g4tools; AnaEx02 – same with ROOT
    - http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples_doc/html/Examples_analysis.html (link)
  - More extended/advanced examples with use of AIDA in Geant4 9.6.p01
    - Now being migrated to g4tools
  - This examples require to be build with an AIDA compliant analysis tools (OpenScientist, osc_batch, ...)

# Using OpenScientist

- Detailed guidance on integrating with Geant4 available at http://openscientist.lal.in2p3.fr



**OpenScientist**

- Introduction
- Download and run
- Build from source
- Build from source (G4)
- The survival kit
- Newcomer ?

- osc-plot
- First AIDA program
- AIDA and Geant4
- Plot customization
- osc-g4-vis
- osc-gui-viewer
- osc-gui-server
- osc-paw
- osc-new-app
- Python
- obuild

- AIDA, G4 under Windows
- AIDA and CERN-ROOT

- Demos
- 3D, 2D viewers
- File formats
- Some code
- Gallery
- License
- Mailing list, support
- Previous versions

- Slides, papers
- FAQ

## Introduction

OpenScientist is an integration of open source products working together to do scientific visualization and data analysis, in particular for high energy physics (HEP).

(OpenScientist is definitely not one million lines of intricated and unnecessary complicated home made code reinventing everything).

### Ask for the programs !

The distribution comes with the osc-plot program that permits to visualize generic common analysis data as histograms, tuples, functions, cloud of points that are stored in file at various formats as plain ascii, aida XML, HDF5, old CERN/hbook and (already old) new CERN/root format. The osc-plot program permits to do interactive manipulations over these kind of data through dedicated GUI (Graphical User Interface) palettes. Someone can do also interactive fitting on them.

From the 16.8, comes also the osc-g4-vis program that permits to instrument in a swift some of the Geant4 examples and then permits to bring high level GUI and visualization to a Geant4 application.

Beside these, there is also the osc-gui-viewer that permits to view a GUI described in files at the XML .onx format. We attempt also to have some compatibility to the HEP past by having the osc-paw program.

### Programming analysis, graphics, GUI

The distribution comes also with an implementation of the AIDA interfaces. These API interfaces permit to easily program some common task needed in batch physics program (for example a simulation) as creating histograms and tuples, fill them and then write then in file at various formats.

But the distribution comes also with the coin3d library that permits to program easily a high level 3D (or not) visualization for your data.

And at last, but not the least, the GUIs of the various applications are described in XML files (at the .onx XML format). The XML files being read at run time, someone can strongly customize the user interface in a very easy and nice way without any C++ programming. OpenScientist comes with various GUI factories that permits, from the XML files, to create GUIs by using OpenMotif, native Win32, Qt, gtk but also SDL. In particular this last one opens the way to "jump" in your data as if being in a video game !

The pages are organised around various tutorials describing how to start quickly with the various programs (mainly osc-plot, osc-g4-vis), how to build simple (and less simple) AIDA and coin3d programs and also how to create some simple GUIs by using the .onx XML format.

- including support for using in Windows

# Conclusions

- Since 9.5 release, Geant4 provides a lightweight analysis tools as part of distribution
- In addition users can choose to use external package functionality in applications' UserActions and link against libraries
- User communities have developed (or adopted) analysis tools suitable to their applications
- Geant4 Analysis Forum
  - http://hypernews.slac.stanford.edu/HyperNews/geant4/get/analysis.html