

User Interface

<http://cern.ch/geant4>

The full set of lecture notes of this Geant4 Course is available at

<http://www.ge.infn.it/geant4/events/nss2004/geant4course.html>

Contents

- Setting up user interfaces
- Analysis
- User commands

Steering the Simulation

- A Geant4 simulation can be steered by a batch session or by commands captured from an interactive session
- Setting up batch mode is easy:
 - In your main()

```
G4UImanager* UI = G4UImanager::GetUIpointer();  
G4String command = “/control/execute”;  
G4String fileName = argv[1];  
UI -> applyCommand(command+fileName);
```
- Setting up interactive mode is also easy – but there are many choices of interface
 - All must be derived from the abstract class G4UISession
 - Geant4 provides several implementations

User Interface Choices

- G4Uterminal – C-shell-like character terminal
 - runs on all Geant4-supported platforms
- G4Ultcsh – tcsh-like character terminal with command completion, history, etc.
 - runs only on Solaris and Linux
- G4UIXm, G4UIXaw, G4UIXWin32 – G4Uterminal implemented over Motif, Athena and WIN32 libraries
 - runs on Unix/linux and Windows, respectively
- G4UIGAG – Java-based GUI
 - runs on all Geant4 platforms

How to Use the Interface

- In your main() add the lines:

```
#include "G4UIxxx.hh" // xxx = terminal, Xm, Xaw, Win32, GAG
G4UISession* session = new G4UIxxx;
UI->ApplyCommand("/control/execute somefile.mac"); // optional
session->SessionStart();
delete session;
```
- For a tcsh session, the second line must be

```
G4UISession* session = new G4UITerminal(new G4UITcsh);
```
- Many examples in
 - `examples/novice/N01, N02, N03,`

Environment Variables

- None required in order to use G4Uterminal, G4Ultcsh, G4UIGAG
 - these sessions do not need external libraries, so they are automatically built and linked
- To build G4UIXm, G4UIXaw, or G4UIXWin32
 - setenv G4UI_USE_XM 1 , or
 - setenv G4UI_USE_XAW 1, or
 - setenv G4UI_USE_WIN32 1

Useful GUI Tools Released by Geant4 Developers

- GGE: Geometry editor based on Java GUI
 - <http://erpc1.naruto-u.ac.jp/~geant4>
- GPE: Physics editor based on Java GUI
 - <http://erpc1.naruto-u.ac.jp/~geant4>
- OpenScientist: interactive environment for analysis
 - <http://www.lal.in2p3.fr/OpenScientist>

Analysis Interface

- Geant4 does not endorse or support particular analysis packages
- However an abstract analysis interface is provided
 - AIDA (Abstract Interfaces for Data Analysis)
 - will be built and linked to your application if environment variable G4ANALYSIS_USE is set
 - AIDA headers must be installed in code where analysis is set up
 - `#include AIDA/AIDA.h`
 - must use AIDA-compliant analysis tools
- More AIDA information:
 - <http://aida.freehep.org>

Some AIDA-compliant Analysis Packages

- Anaphe
 - <http://cern.ch/anaphe/>
- Open Scientist Lab
 - <http://www.lal.in2p3.fr/OpenScientist>
 - an example using Open Scientist can be found in <examples/extended/analysis/AnaEx01>
- JAS (Java Analysis Studio)
 - <http://jas.freehep.org>
 - for analysis example using JAS/AIDA see <examples/extended/analysis/A01>

Built-in User Commands

- Geant4 provides a number of user interface commands which can be used:
 - interactively via a (G)UI
 - in a macro file via `/control/execute <command>`
 - within C++ code using the `ApplyCommand` method of `G4UImanager`
- A complete list of built-in commands, roughly organized according to Geant4 categories is available in the Geant4 Application Developers Guide, Chapter 7.1

User-defined Commands

- If built-in commands are not enough, you can make your own
- Geant4 provides several command classes, all derived from G4UIcommand:
 - G4UIcmdWithoutParameter
 - G4UIcmdWithABool
 - G4UIcmdWithADouble
 - and many more
- Example

```
energyCmd = new G4UIcmdWithADoubleAndUnit("/gun/energy", this);
energyCmd->SetParameterName("Energy",true,true);
energyCmd->SetDefaultUnit(GeV);
```
- See also Application Developers Guide, Chapter 7.2