Slides from Joseph Perl (SLAC) and Laurent Garnier (LAL/IN2P3)
Tutorials

And here is a good time to discuss graphics (not bitmapped graphics). After using the viewer's zoom feature, the image will...

Now that you have some basic functionality, create a file called: geant4/examples/extended/analysis/AnalysisA switched on and off.

One last command for the detector was drawn:

The output appears as included automatically.

To select from among right mouse button (or)

Orientation Actions

Or, if you skipped running Geant4 for

Run DAWN on this file:

dawn g4_01.prim

Go to the DAWN GUI's page 1 and

On the same page, change "Camera Position" to some

You should end up with an image at

Second tutorial: User Interface demo

Step 5: Qt interface

Step 6: Making movies (at 64'12"

Step 7: Export high quality pictures (at 6'12"

/vis/viewer/exec/run/beamon 1

The detector will then...

to still see the tracks

X viewer-0 (OpenGL)
Simulation data can be visualized:

- Geometrical components
- Particle trajectories and tracking steps
- Hits of particles in the geometry
- Scored energy, dose, etc.

Other user defined objects can be visualized:

- Polylines
  - such as coordinate axes
- 3D Markers
  - such as eye guides
- Text
  - descriptive character strings
  - comments or titles
## Choices

<table>
<thead>
<tr>
<th>Driver</th>
<th>Variant</th>
<th>High quality print</th>
<th>Interactive</th>
<th>browse geometry hierarchies</th>
<th>Direct access to G4 kernel</th>
<th>Make movies</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenGL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Xm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Qt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Win32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenInventor</td>
<td>Xt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Win32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAWN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HepRep</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gMocren</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RayTracer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACSII File</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You may ask why Geant4 has so many different visualization systems.
This is a natural result of Geant4 being a toolkit and not a single application.

To support user communities who incorporate Geant4 into their own pre-existing software frameworks, Geant4 visualization is built around a set of well defined interfaces.

- These interfaces make it straightforward to connect Geant4's core visualization tools to any visualization system
  - able to drive advanced systems that can natively display complex solids such as Geant4’s cut cylinders
  - able to drive more basic systems that do not understand such solids (system can ask Geant4 visualization to deconstruct complex solids into simpler polygons)
- For those users who want a ready-made visualization solution from Geant4, these same interfaces have made it straightforward for us to provide a variety of solutions, each with particular areas of strength.

Interfaces discussed in detail in: [The Geant4 Visualisation System](#)  
Controlling Which Drivers are Available

Six of the visualization drivers are always included by default (since they require no external libraries):

- HepRepFile
- DAWNFILE
- VRMLFILE
- RayTracer
- gMocrenFile
- ASCIIFile

Other visualization drivers are included only if you request them in your cmake options.

You can also add your own visualization driver:

- Geant4's visualization system is modular: By creating just three new classes, you can direct Geant4 information to your own visualization system.
Simplest command example

- Visualize your geometry in OpenGL
  - /vis/open OGL
  - /vis/drawVolume

- Most examples come with a visualization macro more complete:
  good starting point
Screenshots on the Visualization drivers

- OpenGL
- OpenInventor
- HepRep
- DAWN
- VRML
- RayTracer
- gMocren
OpenGL

- /vis/open OGL

- Features
  - Control directly from Geant4
  - Uses GL libraries that are already included on most Linux and Windows systems
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Fast response (can usually exploit full potential of graphics hardware)
  - Save as pixel graphics or vector EPS
  - Live movies
OpenInventor

- /vis/open OIX or /vis/open IOWin32

- Features
  - Control from the OpenInventor GUI
  - Requires addition of OpenInventor libraries (freely available for most Linux systems and Windows).
  - Rendered, photorealistic image
  - Many interactive features
    - zoom, rotate, translate
    - click to “see inside” opaque volumes
    - click to show attributes (momentum, etc., dumps to standard output)
  - Fast response (can usually exploit full potential of graphics hardware)
  - Expanded printing ability (vector and pixel graphics)

Warning: OpenScientist (implementing our OI driver) is discontinued, but you could still try to use it
OpenInventor: Start from Geant4

- With OpenInventor, start from Geant4, but then some control from OpenInventor GUI
HepRep

- /vis/open HepRepFile

Features

- Create a file to view in the
  - HepRAApp HepRep Browser
  - WIRED4 JAS Plugin
  - or FRED Event Display
- Requires one of the above browsers (freely available for all systems)
- Wireframe or simple area fills (not photorealistic)
- Many interactive features
  - zoom, rotate, translate
  - click to show attributes (momentum, etc.)
  - special projections (FishEye, etc.)
  - control visibility from hierarchical (tree) view of data
- Hierarchical view of the geometry
- HepRAApp and WIRED4 can export to many vector graphic formats (PostScript, PDF, etc.)

Warning: Issues with recent java versions being worked out
HepRAApp: Shows Geometry Hierarchy
DAWN

- /vis/open DAWNFILE

Features
- Create a .prim file
- Requires DAWN, available for all Linux and Windows systems
- DAWN creates a rendered, photorealistic PostScript image
- No interactive features once at PostScript stage
- Highest quality technical rendering - vector PostScript
- View or print from your favorite PostScript application
DAWN Examples

From a repository of images at

DAWNCUT and DAVID

A standalone program, DAWNCUT, can perform a planar cut on a DAWN image.

- DAWNCUT takes as input a .prim file and some cut parameters. Its output is a new .prim file to which the cut has been applied.

Another standalone program, DAVID, can show you any volume overlap errors in your geometry.

- DAVID takes as input a .prim file and outputs a new .prim file in which overlapping volumes have been highlighted.

Details at http://geant4.kek.jp/~tanaka/
VRML

- /vis/open VRML1FILE or /vis/open VRML2FILE

- **Features**
  - Create a file to view in any VRML browser (some as web browser plug-ins).
  - Requires VRML browser (many different choices for different operating systems).
  - Rendered, photorealistic image with some interactive features
    - zoom, rotate, translate
  - Limited printing ability (pixel graphics, not vector graphics)
VRML
- Geant4 creates VRML File
  - /vis/open VRML1FILE or /vis/open VRML2FILE
- View file in a VRML Browser
  - Many free options, for example, here is one from freeWRL
RayTracer

- /vis/open RayTracer

- **Features**
  - Create a jpeg file (and with RayTracerX option, also draws to x window)
  - Forms image by using Geant4's own tracking to follow photons through the detector
  - Can show geometry but not trajectories
  - Can render any geometry that Geant4 can handle (such as Boolean solids) - no other Vis driver can handle every case
  - Supports shadows, transparency and mirrored surfaces
RayTracer works by using Geant4's own tracking to shoot photons through the detector onto a sensitive plane. The resulting image is presented as a jpeg file.

- /vis/open RayTracer

Some pieces of geometries may fail to show up in other visualization drivers (due to algorithms those drivers use to compute visualizable shapes and polygons), but RayTracer can handle any geometry that the Geant4 navigator can handle.
RayTracer can not be used to visualize Trajectories.

Commands:

- 1) trace * Start the ray tracing.
- 2) column * Define the number of horizontal pixels.
- 3) row * Define the number of vertical pixels.
- 4) target * Define the center position of the target.
- 5) eyePosition * Define the eye position.
- 6) lightDirection * Define the direction of illumination light.
- 7) span * Define the angle per 100 pixels.
- 8) headAngle * Define the head direction.
- 9) attenuation * Define the attenuation length for transparent material.
- 10) distortion * Distortion effect of the fish eye lens.
- 11) ignoreTransparency * Ignore transparency even if the alpha of G4Colour < 1
- 12) backgroundColour * Set background colour: red green blue: range 0.->1.
RayTracer Shows Shadows
RayTracer Handles Boolean Solids
Great tool available for volume visualization

- From JST/CREST project (Japan) to improve Geant4 for medical physics

- Able to visualize:
  - Volume data (including overlay of more than one set)
  - Trajectories
  - Geometry

- Runs on:
  - Windows and Linux
  - Mac will likely happen soon
  - Based on a commercial package but offered freely to all Geant4 users
  - [http://geant4.kek.jp/gMocren](http://geant4.kek.jp/gMocren)
  - Installation is straightforward, follow the Download link on the above page
    - First run gMocren’s one-click installer
    - Then, inside C:\Program Files\gMocren\gtk, you will find the one-click installer for gtk
Qt Driver (with OpenGL visualization)

- Recent developments focused on Qt User Interface and Visualization
- Demo…
Interactive commands to G4 kernel

Drift Chamber 2 has 5 hits.
Layer[2]: time 34.70610091825 (nsec) --- local (x,y) = -224.166605113171, -0.21353242280892
Layer[1]: time 36.37649752914 (nsec) --- local (x,y) = -251.45832124629, -1.23342831320323
Layer[2]: time 38.04669415875 (nsec) --- local (x,y) = -278.616463682, -2.20654349158955
Layer[3]: time 39.71701861278 (nsec) --- local (x,y) = -306.0356668968, -3.15098976126989
Layer[4]: time 41.380128111728 (nsec) --- local (x,y) = -333.34494462692, -4.22353375131901

EM Calorimeter has 7 hits. Total Edep is 967.12227158091 (MeV)
Hadron Calorimeter has 0 hits. Total Edep is 0 (MeV)

WARNING: I event has been kept for refreshing and/or reviewing.
"/vis/revlewdeep/Events" to review them.
"/control/shell is
Output from G4 kernel (support for search, MT)

Drift Chamber 2 has 5 hits.
Layer[0]: time 34.70610916825 (nsec) -- local (x,y) = -224.16660513171, 0.21355242280892
Layer[1]: time 36.3964097528104 (nsec) -- local (x,y) = -251.45812124529, -1.2313428323023
Layer[2]: time 38.06626157875 (nsec) -- local (x,y) = -278.616463382, -2.2055434910955
Layer[3]: time 39.7717018412775 (nsec) -- local (x,y) = 306.0156668968, -3.1509879612698
Layer[4]: time 41.380129811728 (nsec) -- local (x,y) = 333.349445269, -4.22231937313901

Hadron Calorimeter has 7 hits. Total Edep is 967.12227135091 (MeV)

Run terminated.

Session: |
Visualization, one tab per viewer

Interactivity with mouse: rotate, zoom, move, pick

Drift Chamber 2 has 5 hits.
Layer[0] : time 34.70610918825 (nsec) --- local (x,y) -224.16660951371, -0.21355242280892
Layer[1] : time 36.37640972814 (nsec) --- local (x,y) -251.45832124829, -1.233428313023
Layer[2] : time 38.64694161873 (nsec) --- local (x,y) -278.6164463582, -2.2065434910955
Layer[4] : time 41.30329911728 (nsec) --- local (x,y) -333.34494462969, -4.2235337313901
EH Calorimeter has 7 hits. Total Edep is 967.12227158091 (MeV)
Hadrion Calorimeter has 0 hits. Total Edep is 0 (MeV)

Run terminated.
Run Summary
Number of events processed : 1
User=0.078 Real=0.099 Sys=0.018
... write Root file /vis/85.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/vis/rev/85events" to review them.
"/control/aball is"
Toolbar and menubar controlled by icons.mac file, add your own without coding.
Help tree browser, free text search

Drift chamber 2 has 5 hits.
Layer[0] : time 34.70610916925 (nsec) --- local (x,y) =-224.16660513711, -0.213552242280892
Layer[1] : time 36.3764579214 (nsec) --- local (x,y) =-251.4982124629, -1.2334283123023
Layer[2] : time 38.0469451775 (nsec) --- local (x,y) =-278.6184435828, -2.265343619555
Layer[4] : time 41.3872911128 (nsec) --- local (x,y) =-333.3649482892, -4.2831337113901
R0: 3.85 cm 9 hits Total Edep is 967.12227138691 (MeV)
R1: 3.85 cm 9 hits Total Edep is 0 (MeV)
Run terminated.
Run Summary
Number of events processed : 1
User=0.06s Real=0.09s Sys=0.01s
... write Root file : ss.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/v4a/reviewOpt2Events" to review them.
/control/shell is

Session : 45
History, re-select command

Drift Chamber 2 has 5 hits.
Layer[0] : time 36.70610996625 (nsec)  ---  local (x,y) = 224.16660521371, -0.21155242280892
Layer[1] : time 36.79610996625 (nsec)  ---  local (x,y) = 251.45932124629, -1.2354263123032
Layer[2] : time 38.04656157875 (nsec)  ---  local (x,y) = -278.05146433828, -2.2054439189255
Layer[4] : time 41.38729211728 (nsec)  ---  local (x,y) = -333.36494482892, -4.2231537511901
Ecal Calorimeter has 7 hits. Total Edep is 967.12227238091 (MeV)
Hadron Calorimeter has 0 hits. Total Edep is 0 (MeV)
Run terminated.
Run Summary
Number of events processed : 1
User=0.07s Real=0.10s Sys=0.01s
... write Root file : S5.root - done
WARNING: 1 event has been kept for refreshing and/or reviewing.
"/data/ rumoursEvents" to review them.
/control/shell ls

Session : 46
Scene tree: edit visualization (e.g. hide volumes)
You can make movies that show Time Development of an event

• I.e., a shower in slow motion

Based on technique of “time-slicing”, breaking trajectories into individual slices, each with a time attribute.

• requires newer visualization features, rich trajectory and some extensions to the OpenGL driver
• you can run these animations Directly from Geant4, does NOT involve stitching together a movie by hand

A collection of example movies has been prepared by John Allison:
http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/

How-To Presentation:
http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.ppt
http://geant4.slac.stanford.edu/Presentations/vis/HowToMakeAMovie.pdf
10 GeV pion

3 ns

Mpeg4 encoding with QuickTime Pro

http://www.hep.man.ac.uk/u/johna/pub/Geant4/Movies/pi-10Gevpi+neutronSideView.mp4
References

• gMocren Home Page http://geant4.kek.jp/gMocren
• DAWN Home Page http://geant4.kek.jp/~tanaka/DAWN/About_DAWN.html
  • DAWNCUT Home Page http://geant4.kek.jp/~tanaka/DAWN/About_DAWNCUT.html
  • DAVID Home Page http://geant4.kek.jp/~tanaka/DAWN/About_DAVID.html
• Satoshi Tanaka’s GEANT4 Ritsumeikan University Group Home Page (more information on DAWN, sample PRIM files, images, etc.) http://geant4.kek.jp/~tanaka/
• HepRAppl HepRep Browser http://www.slac.stanford.edu/~perl/HepRAppl
• OpenScientist Home Page http://openscientist.lal.in2p3.fr
• Under Development: Wt driver, support for browser based visualization (e.g. visualization in a dynamic webpage)