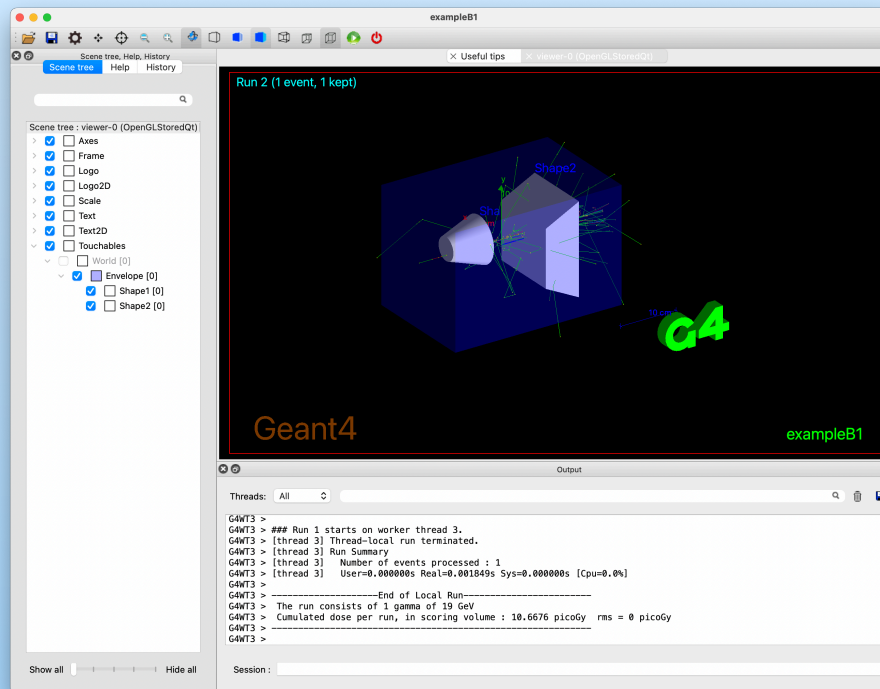


# Introduction to Qt driver



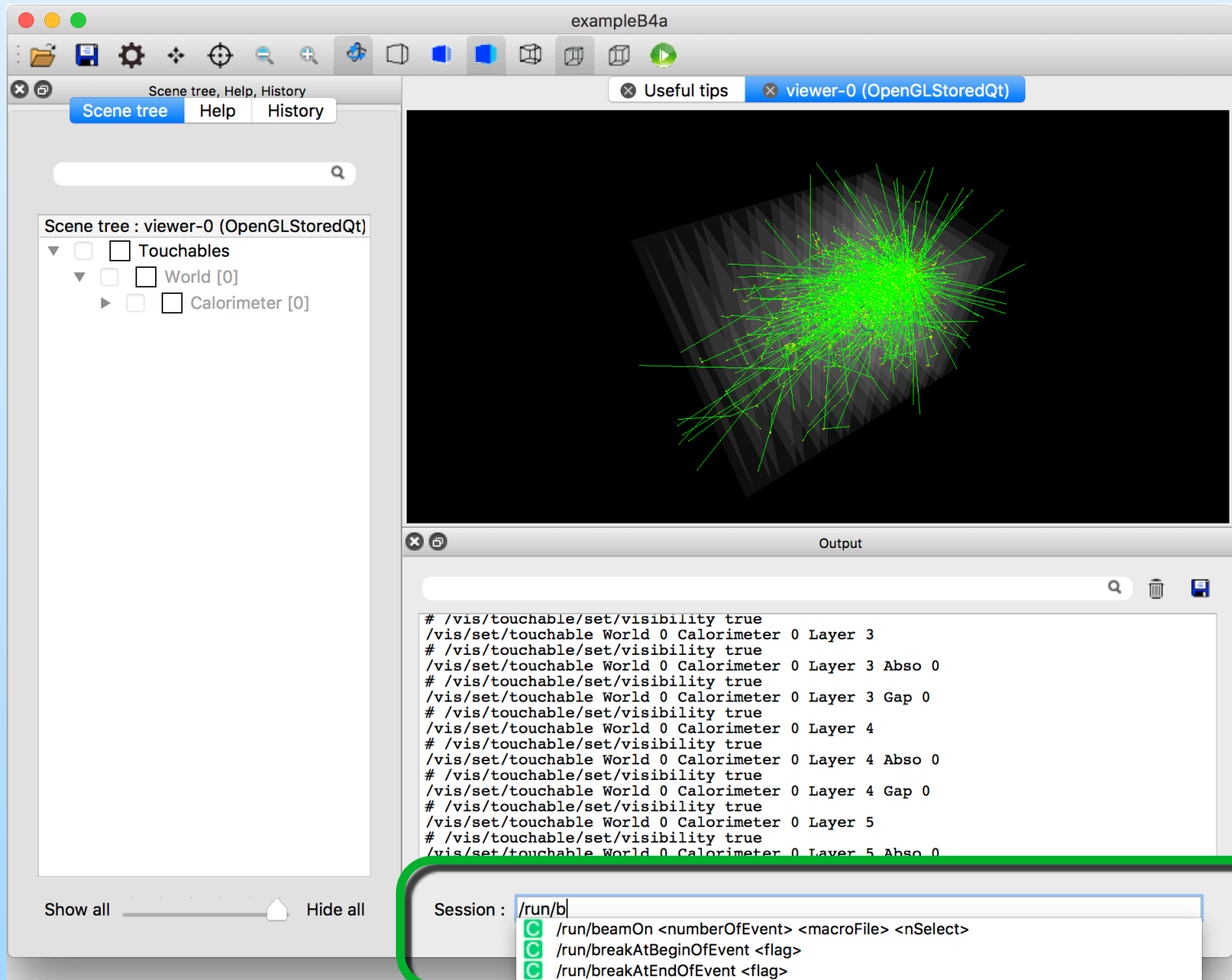
Igor Semeniouk  
LLR, CNRS - Ecole Polytechnique

Slides from Laurent GARNIER  
IRISA / INS2I / CNRS



# Qt driver - Command line

« Tab » is a useful way to complete a command



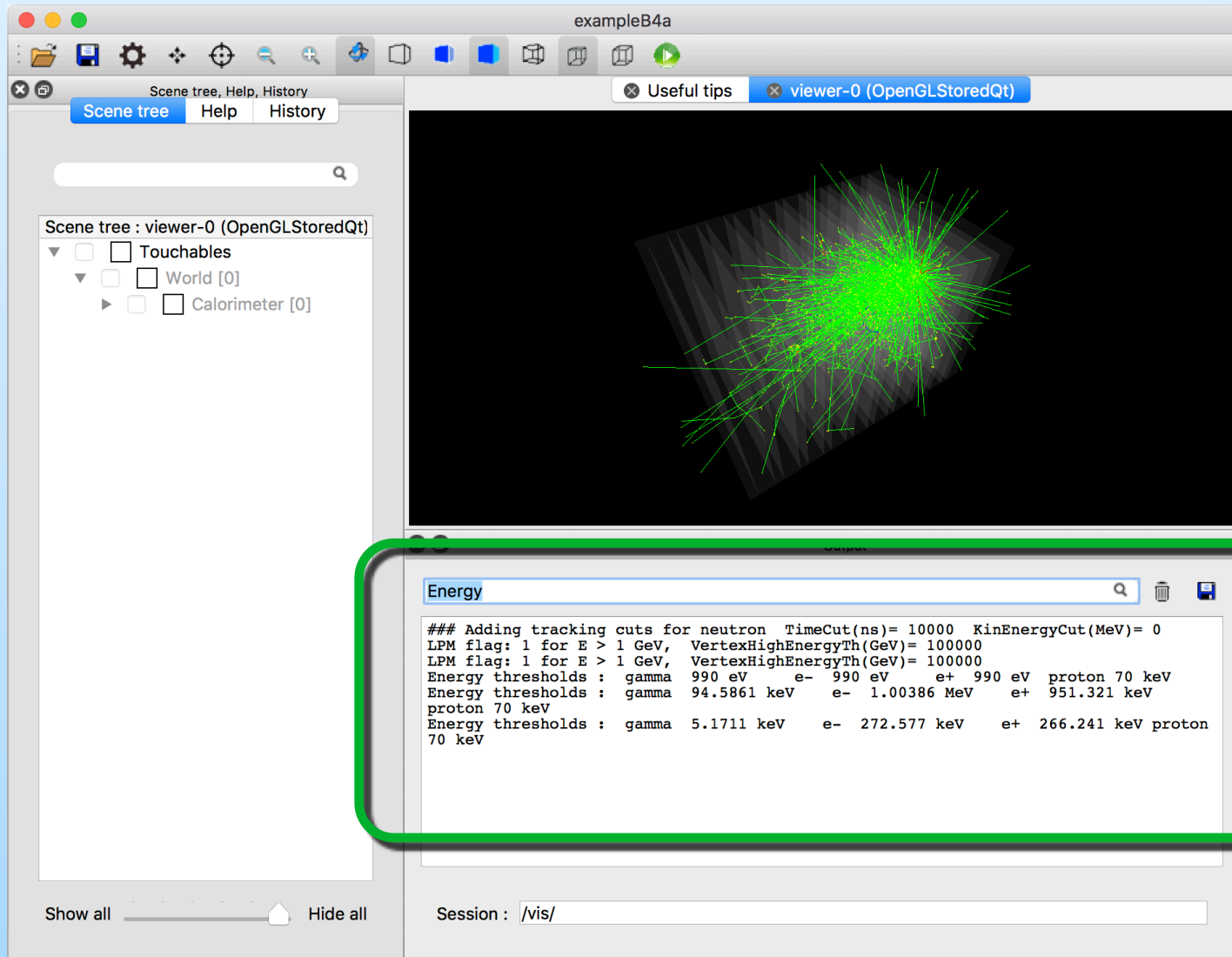
The screenshot displays the Qt driver interface for a 3D visualization. The window title is "exampleB4a". The interface is divided into several panels:

- Scene tree:** Located on the left, it shows a hierarchical view of the scene. The tree structure is:
  - Touchables
    - World [0]
      - Calorimeter [0]

- 3D Viewport:** The central area shows a 3D visualization of a calorimeter detector. It features a central cluster of green points and lines, representing particle tracks, set against a dark background with a grid.
- Output:** A text area at the bottom right displays the command line output. The output consists of several lines of configuration commands for the calorimeter's visibility and layer settings.
- Command Line:** At the bottom, a text input field shows the current session command: `Session : /run/b`. A dropdown menu is open, showing the following options:
- `/run/beamOn <numberOfEvent> <macroFile> <nSelect>`
- `/run/breakAtBeginOfEvent <flag>`
- `/run/breakAtEndOfEvent <flag>`

# Qt driver - Output

See all outputs, you can add filter



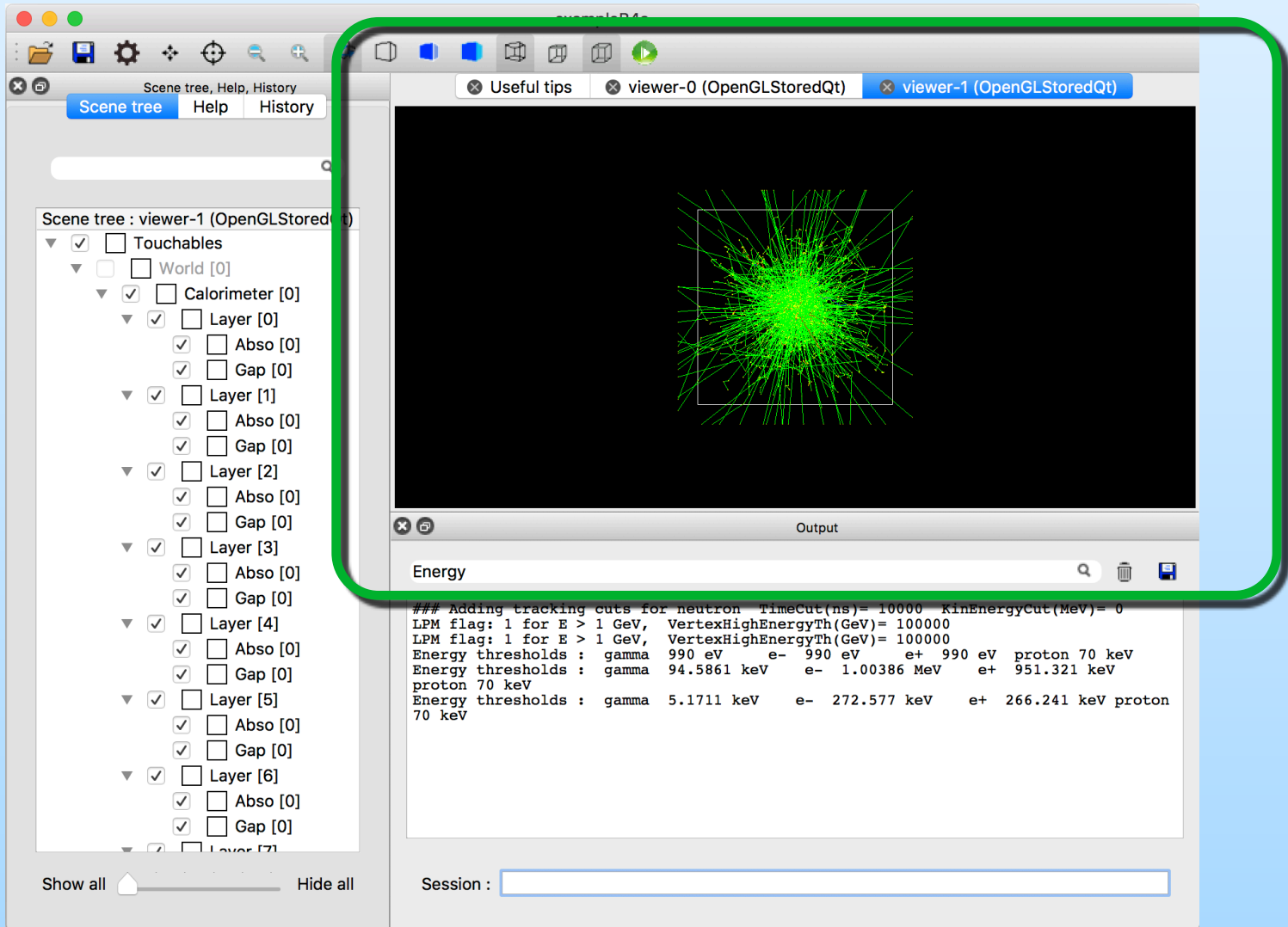
The screenshot displays the Qt driver interface for a simulation. The main window, titled "exampleB4a", features a toolbar with various navigation and visualization tools. On the left, a "Scene tree" panel shows a hierarchical view of the scene, including "Touchable", "World [0]", and "Calorimeter [0]". The central area is a 3D viewer showing a dense cluster of green particle tracks originating from a central point, with a semi-transparent rectangular plane intersecting them. Below the viewer, a terminal window titled "Energy" is highlighted with a green border, displaying the following configuration text:

```
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV
proton 70 keV
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton
70 keV
```

At the bottom of the interface, there is a "Session" field containing the text "/vis/".

# Qt driver - Visualization

Visualization window, one tab by viewer



# Qt driver - Toolbar

Toolbar is controlled by default icons. You can add your own icons by providing a macro file

The screenshot displays the Qt driver interface. At the top, a toolbar is highlighted with a green oval, containing various icons for file operations, settings, navigation, and visualization. Below the toolbar, the interface is divided into several panels:

- Scene tree:** A hierarchical tree view on the left side, showing the structure of the scene. It includes a search bar and a 'Show all' / 'Hide all' toggle at the bottom. The tree is expanded to show a 'Calorimeter [0]' object, which contains multiple 'Layer' objects (Layer [0] through Layer [7]). Each layer contains 'Abso [0]' and 'Gap [0]' objects.
- Viewer:** A central 3D visualization window showing a complex, starburst-like structure of green lines and points, representing the calorimeter's internal structure.
- Output:** A text-based output window at the bottom right, displaying configuration parameters and energy thresholds. The output text is as follows:

```
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV
proton 70 keV
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton
70 keV
```

# Qt driver - Help

Help tree browser, you can search for a word inside command help

The screenshot shows the Qt driver interface. On the left, a help browser is open, displaying search results for the word "export". The search results table is as follows:

Command	Match
vis/viewer/interpolate	
vis/ogl/set/exportFormat	
vis/ogl/export	

Below the search results, there is a "Guidance" section for the "vis/viewer/interpolate" command:

**Guidance :** Interpolate views defined by the first argument, which can contain Unix-shell-style pattern matching characters such as '\*', '?' and '[' - see "man sh" and look for "Pattern Matching". The contents of each file are assumed to be "/vis/viewer" commands that specify a particular view. The files are processed in alphanumeric order of filename. The files may be written by hand or produced by the "/vis/viewer/save" command. The default is to search the working directory for files with a .q4view extension. Another

Below the guidance, there is a table of parameters:

Parameter	Guidance	Type	Ommitable	D
1 pattern	Pattern that defines the view files.	s	True	*.q4
2 no-of-points	Number of interpol...	i	True	50
3 wait-time	Wait time per interpol...	s	True	20.
4 time-unit		s	True	mill
5 export		s	True	no

On the right, the main window displays a 3D visualization of particle tracks, showing a dense cluster of green lines radiating from a central point. Below the visualization, there is an "Output" window showing the following text:

```
Energy
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV
proton 70 keV
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton
70 keV
```

At the bottom, there is a "Session" input field.

# Qt driver - History

History : Double clic on a item to send it to Command line session

The screenshot displays the Qt driver interface for 'exampleB4a'. The interface is divided into several panels:

- History Panel (left, highlighted with a green border):** Contains a list of commands: `/run/beamOn 100` and `/vis/open OGL`. The 'History' tab is selected.
- 3D Visualization (center):** Shows a complex, starburst-like structure of green lines and points, representing a particle simulation or detector response.
- Output Panel (bottom):** Displays the following text:

```
Energy  
## Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0  
LPFlag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000  
LPFlag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000  
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV  
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV  
proton 70 keV  
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton  
70 keV
```
- Session Input (bottom):** A text box labeled 'Session : ' for entering commands.

# Qt driver - Volume tree

Scene tree : you can set visible/unvisible, change color or transparency on volumes

The screenshot displays the Qt driver interface for a detector simulation. The main window, titled 'exampleB4a', contains a 3D viewer showing a detector geometry with a central interaction point. A dense cluster of green lines represents particle tracks originating from this point and passing through various layers of the detector. The detector is composed of several layers, each with an absorber (Abso) and a gap (Gap). The layers are color-coded: Layer 0 (red), Layer 1 (green), Layer 2 (yellow), and Layer 3 (blue). The 3D viewer also shows a grey rectangular block on the right side.

The 'Scene tree' panel on the left provides a hierarchical view of the scene. It is titled 'Scene tree : viewer-0 (OpenGLStoredQt)'. The tree structure is as follows:

- Touchable
  - World [0]
    - Calorimeter [0]
      - Layer [0]
        - Abso [0] (checked, red)
        - Gap [0] (checked, green)
      - Layer [1]
        - Abso [0] (checked, red)
        - Gap [0] (checked, green)
      - Layer [2]
        - Abso [0] (checked, yellow)
        - Gap [0] (checked, white)
      - Layer [3] (checked, blue)
      - Layer [4]
        - Abso [0] (unchecked)
        - Gap [0] (unchecked)
      - Layer [5]
        - Abso [0] (unchecked)
        - Gap [0] (unchecked)
      - Layer [6]
        - Abso [0] (unchecked)
        - Gap [0] (unchecked)
      - Layer [7]
        - Abso [0] (unchecked)
        - Gap [0] (unchecked)

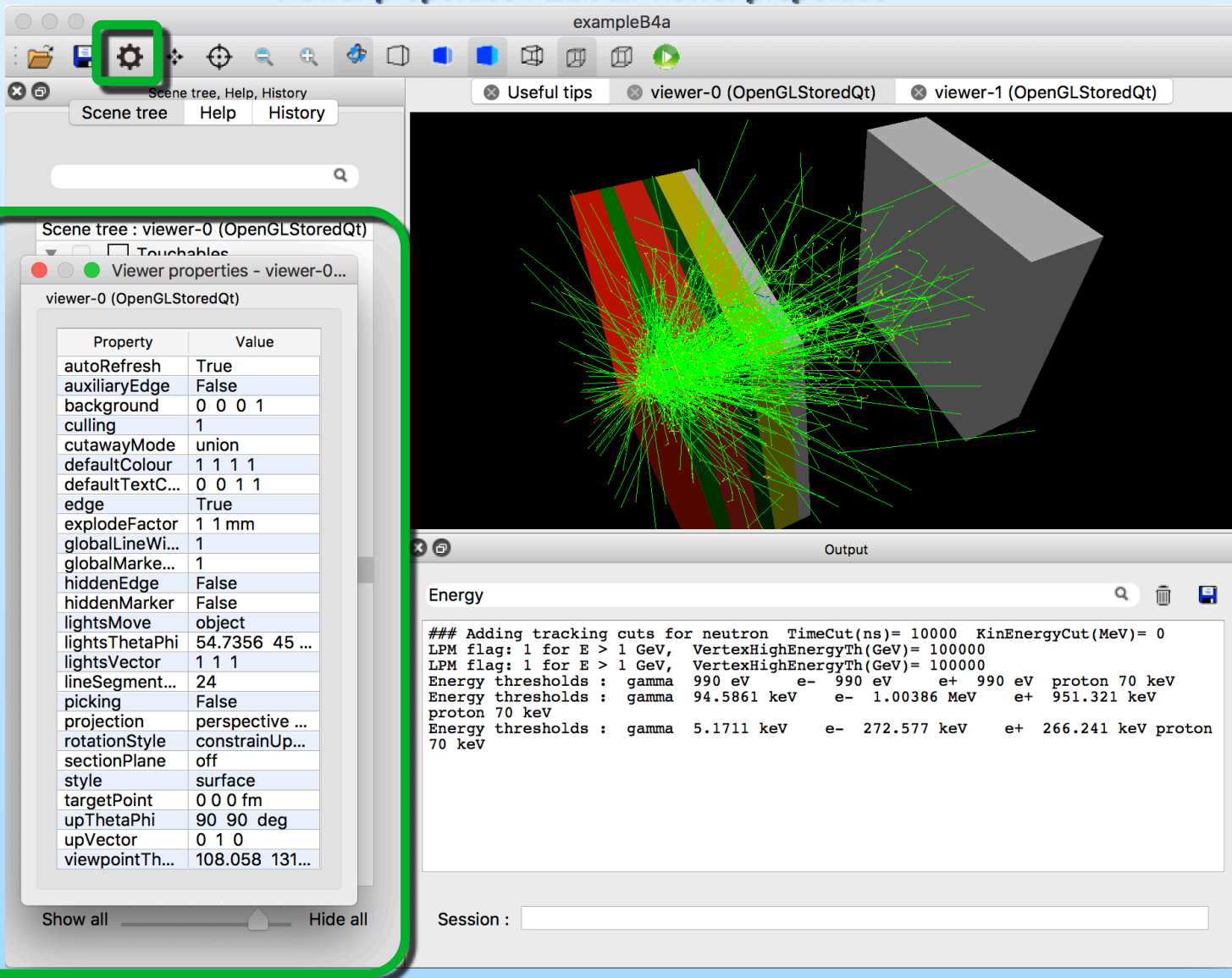
A tooltip 'double-click to change the color' is visible over the 'Calorimeter [0]' node. The 'Output' panel at the bottom shows the following text:

```
Energy
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV
proton 70 keV
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV
proton 70 keV
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton
70 keV
```



# Qt driver - Volume tree

## Viewer properties : Edit all viewer properties



The screenshot displays the Qt driver interface for a Geant4 simulation. The main window, titled "exampleB4a", shows a 3D visualization of particle tracks in a volume. The tracks are represented by a dense cluster of green lines originating from a central point and extending outwards, interacting with a grey rectangular volume. The background is black, and the tracks are colored in a gradient from red to yellow to green.

On the left side, the "Scene tree" panel is visible, showing the hierarchy of objects in the scene. The "Viewer properties - viewer-0..." dialog is open, displaying a list of properties and their values for the selected viewer. The properties are:

Property	Value
autoRefresh	True
auxiliaryEdge	False
background	0 0 0 1
culling	1
cutawayMode	union
defaultColour	1 1 1 1
defaultTextC...	0 0 1 1
edge	True
explodeFactor	1 1 mm
globalLineWi...	1
globalMarke...	1
hiddenEdge	False
hiddenMarker	False
lightsMove	object
lightsThetaPhi	54.7356 45 ...
lightsVector	1 1 1
lineSegmen...	24
picking	False
projection	perspective ...
rotationStyle	constrainUp...
sectionPlane	off
style	surface
targetPoint	0 0 0 fm
upThetaPhi	90 90 deg
upVector	0 1 0
viewpointTh...	108.058 131...

Below the properties table, there are "Show all" and "Hide all" buttons. The "Output" panel at the bottom right shows the following text:

```
Energy  
### Adding tracking cuts for neutron TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0  
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000  
LPM flag: 1 for E > 1 GeV, VertexHighEnergyTh(GeV)= 100000  
Energy thresholds : gamma 990 eV e- 990 eV e+ 990 eV proton 70 keV  
Energy thresholds : gamma 94.5861 keV e- 1.00386 MeV e+ 951.321 keV  
proton 70 keV  
Energy thresholds : gamma 5.1711 keV e- 272.577 keV e+ 266.241 keV proton  
70 keV
```

The "Session" field at the bottom is empty.

# Qt driver - Volume tree

Picking widget: Pick a scene object and see all its properties

The screenshot displays the Qt driver interface for Geant4. The main window, titled 'exampleB4a', contains a 3D visualization of a detector volume with green particle tracks. A green box highlights the 'Picking widget' icon in the toolbar. A 'Scene tree' panel on the left shows a hierarchical list of objects, with the selected object highlighted in green. An 'Output' window at the bottom right displays the properties of the selected object.

**Scene tree:** viewer-0 (OpenGLStoredQt)  
8 objects selected - viewer-0 (OpenGLStoredQt)

- Volume: World:0/Calorimeter:0/Layer:7/Gap:0
- Volume: World:0/Calorimeter:0/Layer:8
- Volume: World:0/Calorimeter:0/Layer:8/Abso:0
- Volume: World:0/Calorimeter:0/Layer:8/Gap:0
- Volume: World:0/Calorimeter:0/Layer:9
- Volume: World:0/Calorimeter:0/Layer:9/Abso:0
- Volume: World:0/Calorimeter:0/Layer:9/Gap:0
- Trajectory: Run: 0, Event: 34

**Output:**

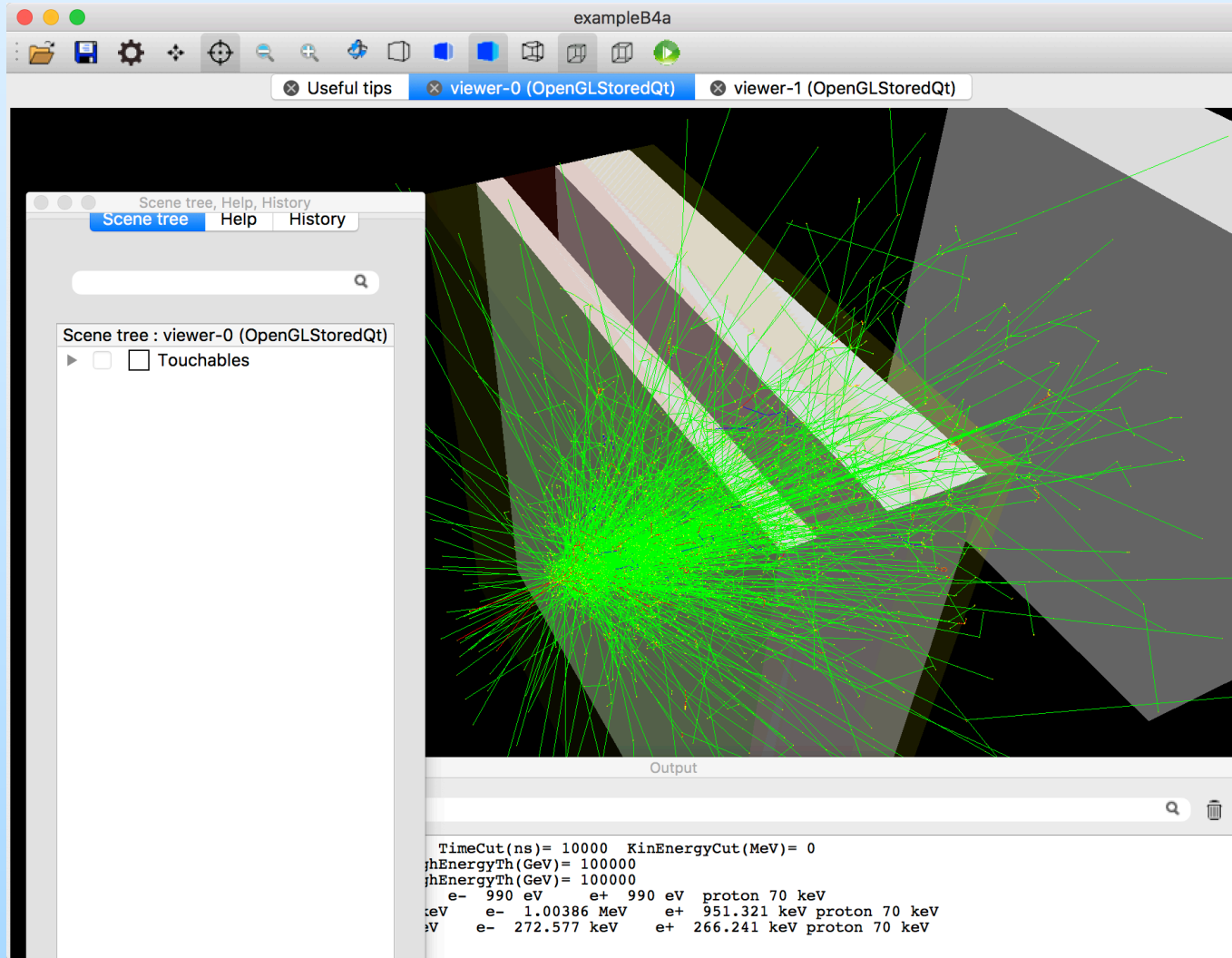
```
TimeCut(ns)= 10000 KinEnergyCut(MeV)= 0
hEnergyTh(GeV)= 100000
hEnergyTh(GeV)= 100000
eV 990 eV e+ 990 eV proton 70 keV
eV e- 1.00386 MeV e+ 951.321 keV
V e- 272.577 keV e+ 266.241 keV proton
```

**Selected Object Properties:**

```
G4TrajectoriesModel:
Run ID (RunID): 0
Event ID (EventID): 34
G4SmoothTrajectory:
Track ID (ID): 28
Parent ID (PID): 8
Particle Name (PN): gamma
Charge (Ch): 0 (e+)
PDG Encoding (PDG): 22
Initial kinetic energy (IKE): 1.97004 MeV (G4BestUnit)
Initial momentum (IMom): 0.220729 1.08047 1.63245 MeV (G4BestUnit)
Initial momentum magnitude (IMag): 1.97004 MeV (G4BestUnit)
No. of points (NTP): 12
G4SmoothTrajectoryPoint:
Step Position (Pos): 0.360103 -1.62146 -1.33489 cm (G4BestUnit)
G4SmoothTrajectoryPoint:
```

# Qt driver - Volume tree

## Dockable widgets



# References

**Geant4 Visualization home page**

[http://geant4-userdoc.web.cern.ch/geant4-userdoc/  
UsersGuides/ForApplicationDeveloper/html/Visualization/  
visualization.html](http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/Visualization/visualization.html)

**Geant4 Qt Home Page**

<http://geant4.in2p3.fr/spip.php?rubrique25&lang=en>