

# New vis (and some GUI) features 2023

Scheduled for Geant4 11.2

# Contents

- G4debug—a new output stream
- A new scene tree—available to all drivers that use the Qt GUI
- /vis/open (without parameters) to get the default driver
  - /vis/open (with parameters), e.g., /vis/open OGL, works as before
  - The default driver can be chosen at run time
    - By programmed argument
    - By environment variable
    - By entry in a file in your home directory, ~/.g4session
  - Otherwise according to batch/interactive and build flags
- New off-screen drivers with ToolsSG (TSG) and Vtk
  - Output to file—choice of formats
  - Any size—choose large size to get high resolution

# G4debug example using Qt GUI

In `B1::SteppingAction::UserSteppingAction` after testing for the scoring volume:

```
G4debug << "Deposited in scorer: "  
<< step->GetTotalEnergyDeposit()/CLHEP::MeV << " MeV" << G4endl;
```

To show occasional debug line in a mass of other output, run with:

```
/tracking/verbose 2  
/gun/particle proton  
/gun/energy 1 GeV  
/run/beamOn
```

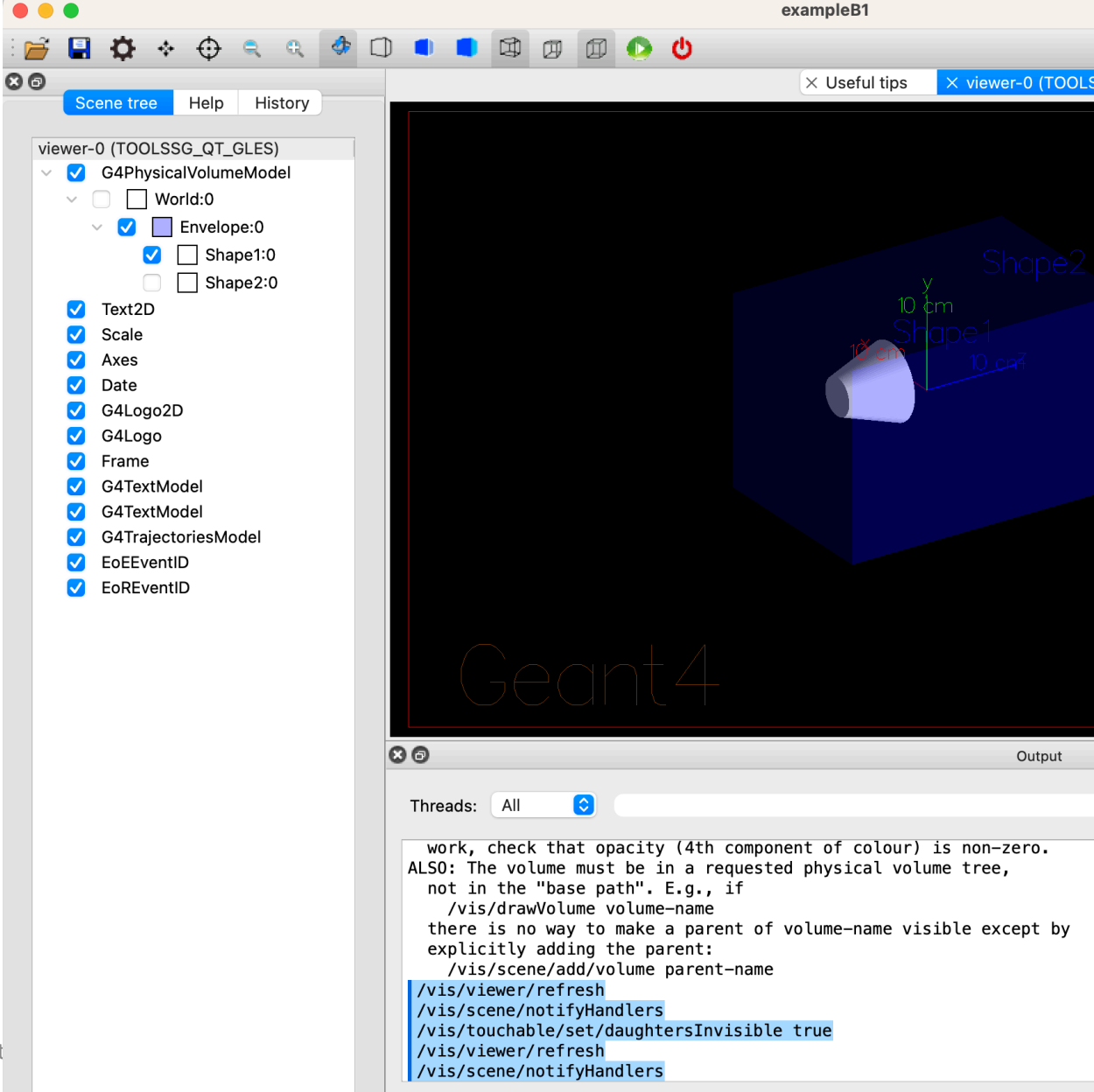
The Qt GUI intercepts this Output stream and highlights it, making it easy to pick out the debug line.  
(Other sessions, e.g., tcsh, simply direct to `std::cout`.)

```
Output  
Threads: All  
G4WT0 > *****  
G4WT0 > * G4Track Information: Particle = pi+, Track ID = 6, Parent ID = 1  
G4WT0 > *****  
G4WT0 >  
G4WT0 > Step#      X          Y          Z          KineE      dEStep     StepLeng   TrakLeng   Volume   Process  
G4WT0 > 0  2.672 mm  1.177 cm  1.413 cm  116.2 MeV  0 eV       0 fm       0 fm       Envelope  initStep  
G4WT0 > 1 -512.7 um  1.891 cm  4 cm      109.9 MeV  6.236 MeV  2.703 cm  2.703 cm  Envelope  Transportation  
G4WT0 > 2 -776.5 um  1.952 cm  4.223 cm  108.5 MeV  1.042 MeV  2.322 mm  2.935 cm  Shape2    hIoni  
G4WT0 >  
:----- List of secondaries -----  
G4WT0 > e-: energy = 427 keV time = 744 ps  
G4WT0 > :-----  
G4WT0 > Deposited in scorer: 1.04163 MeV  
G4WT0 > 3 -5.445 mm  2.949 cm  7.72 cm  93.64 MeV  14.84 MeV  3.669 cm  6.604 cm  Shape2    hIoni  
G4WT0 > Deposited in scorer: 14.8355 MeV  
G4WT0 > 4 -6.27 mm  3.104 cm  8.217 cm  90.46 MeV  2.02 MeV  5.272 mm  7.131 cm  Shape2    hIoni  
G4WT0 >  
:----- List of secondaries -----  
G4WT0 > e-: energy = 1.159 MeV time = 913.9 ps  
G4WT0 > :-----  
G4WT0 > Deposited in scorer: 2.02035 MeV  
G4WT0 > 5 -9.099 mm  3.686 cm  10 cm    82.03 MeV  8.431 MeV  1.897 cm  9.028 cm  Shape2    Transportation  
G4WT0 > Deposited in scorer: 8.43107 MeV  
G4WT0 > 6 -1.117 cm  4.011 cm  11.23 cm  77.68 MeV  3.295 MeV  1.286 cm  10.31 cm  Envelope  hIoni  
G4WT0 >  
:----- List of secondaries -----  
G4WT0 > e-: energy = 1.010 MeV time = 1.010 ps  
G4WT0 > :-----  
Session: 
```



# New scene tree (contd)

Click on blue check box to make invisible/visible



The screenshot shows a Geant4 visualization window titled "exampleB1". The window is divided into three main sections:

- Scene tree:** A hierarchical tree view on the left side. The root node is "viewer-0 (TOOLSSG\_QT\_GLES)". It contains several sub-nodes, each with a blue checkmark indicating it is visible. The visible nodes are: G4PhysicalVolumeModel, Envelope:0, Shape1:0, Text2D, Scale, Axes, Date, G4Logo2D, G4Logo, Frame, G4TextModel, G4TextModel, G4TrajectoriesModel, EoEEventID, and EoREventID. Other nodes like "World:0", "Shape2:0", and "Shape2:1" are unchecked, indicating they are invisible.
- 3D Viewport:** A central area showing a 3D visualization of a detector component. The component is a blue, cylindrical-like structure with a central hole. It is positioned on a dark blue base. A coordinate system is visible with axes labeled "x", "y", and "z". Dimensions are indicated: "10 cm" for the height of the component, "10 cm" for the radius of the hole, and "10 cm" for the radius of the base. Labels "Shape1" and "Shape2" are placed near the component.
- Output Console:** A bottom section showing the output of the visualization. It includes a "Threads:" dropdown set to "All" and a text area containing the following text:

```
work, check that opacity (4th component of colour) is non-zero.  
ALSO: The volume must be in a requested physical volume tree,  
not in the "base path". E.g., if  
/vis/drawVolume volume-name  
there is no way to make a parent of volume-name visible except by  
explicitly adding the parent:  
/vis/scene/add/volume parent-name  
/vis/viewer/refresh  
/vis/scene/notifyHandlers  
/vis/touchable/set/daughtersInvisible true  
/vis/viewer/refresh  
/vis/scene/notifyHandlers
```

# New scene tree (contd)

Click on chevron to hide/expose list of daughters in the scene tree

The screenshot shows the Geant4 GUI interface. On the left is the 'Scene tree' panel for 'viewer-0 (TOOLSSG\_QT\_GLES)'. It contains a hierarchical list of objects with checkboxes and chevrons. The 'G4PhysicalVolumeModel' is expanded, showing 'World:0' and 'Envelope:0'. Below these are various models like 'Text2D', 'Scale', 'Axes', 'Date', 'G4Logo2D', 'G4Logo', 'Frame', 'G4TextModel', 'G4TrajectoriesModel', 'EoEEventID', and 'EoREventID'. On the right is the 3D visualization window, which displays a blue cylinder on a dark blue plane. The cylinder is labeled 'Shape1' and has dimensions '10 cm' and '10 cm' indicated. Another label 'Shape2' is visible in the background. The 'Geant4' logo is at the bottom of the visualization window. Below the visualization is an 'Output' window showing a list of threads and a block of text with some lines highlighted in blue.

```
viewer-0 (TOOLSSG_QT_GLES)
├── [x] G4PhysicalVolumeModel
│   ├── [x] World:0
│   │   └── [x] Envelope:0
│   └── [x] [ ]
├── [x] Text2D
├── [x] Scale
├── [x] Axes
├── [x] Date
├── [x] G4Logo2D
├── [x] G4Logo
├── [x] Frame
├── [x] G4TextModel
├── [x] G4TextModel
├── [x] G4TrajectoriesModel
├── [x] EoEEventID
└── [x] EoREventID
```

work, check that opacity (4th component of colour) is non-zero.  
ALSO: The volume must be in a requested physical volume tree,  
not in the "base path". E.g., if  
/vis/drawVolume volume-name  
there is no way to make a parent of volume-name visible except by  
explicitly adding the parent:  
/vis/scene/add/volume parent-name  
/vis/viewer/refresh  
/vis/scene/notifyHandlers  
/vis/touchable/set/daughtersInvisible true  
/vis/viewer/refresh  
/vis/scene/notifyHandlers

# New scene tree (contd)

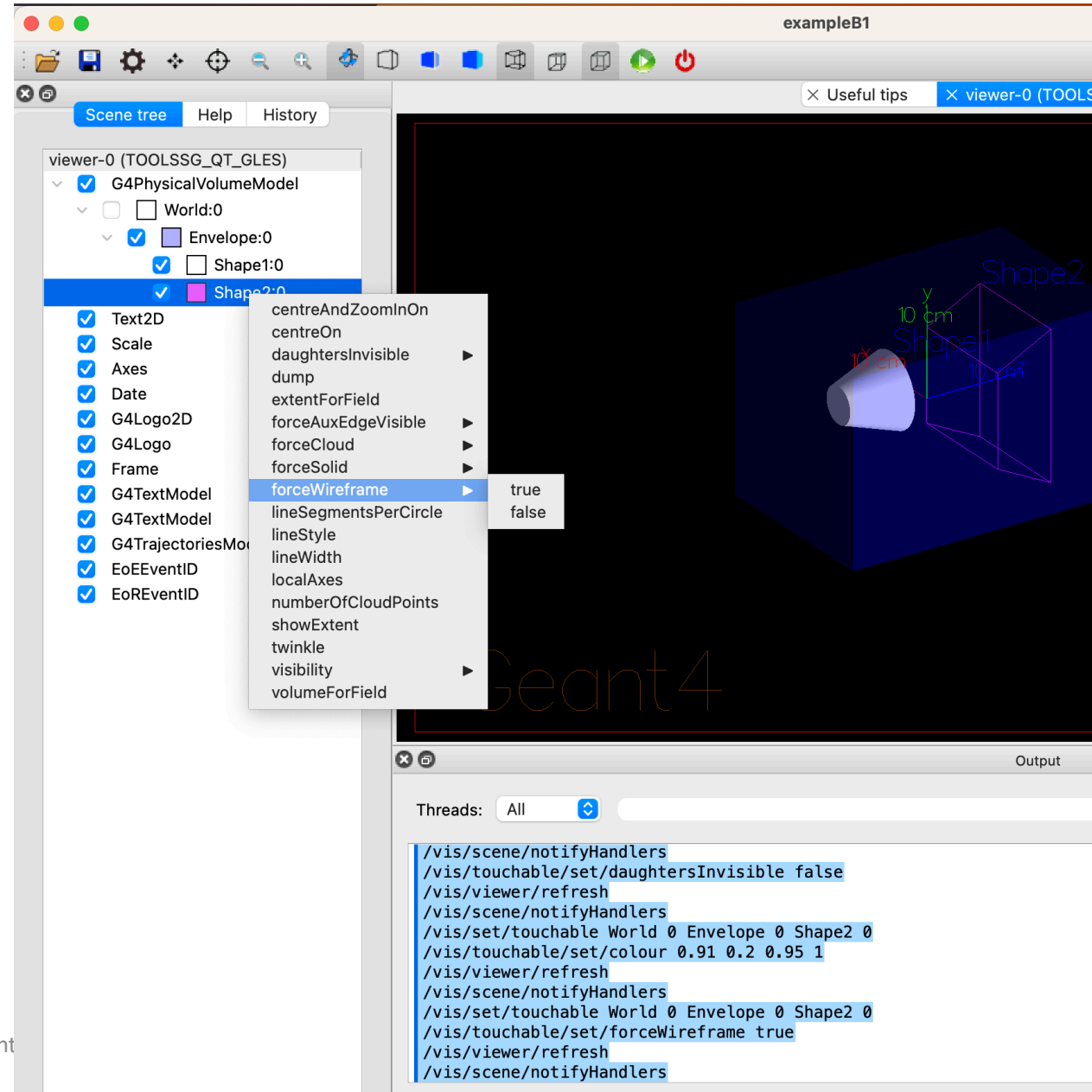
Double click on small square box to change colour

The screenshot displays the Geant4 graphical user interface. On the left, the 'Scene tree' panel shows a hierarchy of objects under 'viewer-0 (TOOLSSG\_QT\_GLES)'. The 'G4PhysicalVolumeModel' is expanded to show 'World:0', which contains 'Envelope:0', 'Shape1:0', and 'Shape2:0'. Below this, a list of other objects is shown with checkboxes, including 'Text2D', 'Scale', 'Axes', 'Date', 'G4Logo2D', 'G4Logo', 'Frame', 'G4TextModel', 'G4TrajectoriesModel', 'EoEEventID', and 'EoREventID'. A 'Colors' dialog box is open over the 3D view, showing a 'Spectrum' color picker and an 'Opacity' slider set to 100%. The 3D view shows a blue cylinder and a purple rectangular prism, with labels 'Shape1' and 'Shape2' and dimensions '10 cm'. The 'Output' console at the bottom shows a series of commands for setting the color of 'Shape2' to (0.91, 0.2, 0.95, 1) and refreshing the view.

```
/vis/scene/notifyHandlers
/vis/set/Touchable World 0 Envelope 0 Shape2 0
/vis/Touchable/set/visibility true
/vis/viewer/refresh
/vis/scene/notifyHandlers
/vis/Touchable/set/daughtersInvisible false
/vis/viewer/refresh
/vis/scene/notifyHandlers
/vis/set/Touchable World 0 Envelope 0 Shape2 0
/vis/Touchable/set/colour 0.91 0.2 0.95 1
/vis/viewer/refresh
/vis/scene/notifyHandlers
```

# New scene tree (contd)

Right-click to get menu of actions





# /vis/open (without parameters)

- This will be the recommended way of opening a viewer
  - Examples B1 and B2 have been changed
  - We plan to change all tests and examples for 11.2
- A new constructor in all main programs:  
`G4VisExecutive(argc, argv)`
- All `/vis/open OGL` commands stripped of parameters in all vis.mac files
- Existing behaviour is preserved
  - The default default is OGL
  - You can still use the old constructor: `G4VisExective()`
  - You can still use `/vis/open OGL`
- The default can be changed at run time (without having to edit vis.mac!)
  - By programmed argument
  - By environment variable
  - By entry in a file in your home directory, `~/.g4session`
- Otherwise according to batch/interactive and build flags

# /vis/open (without parameters) (contd)

In your main program

```
// Initialize visualization with the default graphics system
auto visManager = new G4VisExecutive(argc, argv);
// Constructors can also take optional arguments:
// - a graphics system of choice, eg. "OpenGL"
// - and a verbosity argument - see /vis/verbose guidance.
// auto visManager = new G4VisExecutive(argc, argv, "OpenGL", "Quiet");
// auto visManager = new G4VisExecutive("Quiet");
visManager->Initialize();
```

# /vis/open (without parameters) (contd)

In B1/vis.mac:

Environment: set with **export** or **setenv**:

```
export G4VIS_DEFAULT_DRIVER=Qt3D
```

Or temporarily on the command line:

```
G4VIS_DEFAULT_DRIVER=Vtk ./exampleB1
G4VIS_DEFAULT_DRIVER="TSG_OFFSCREEN 2000x2000" \
./exampleB1
```

~/g4session:

```
Qt # Default session
#exampleB1 tcsh
exampleB1 Qt TSG 1000x1000+0-0
```

In all other vis.mac files:

```
# Open a viewer
/vis/open
# This opens the default viewer - see examples/basic/B1/vis.mac for a
# more comprehensive overview of options. Also the documentation.
```

```
# Specify a viewer, e.g., /vis/open OGL, or allow a system choice:
/vis/open
# This chooses a graphics system (in order of priority):
# - by argument in G4VisExecutive construction.
# - by environment variable, G4VIS_DEFAULT_DRIVER.
# - by information in ~/.g4session.
# - by mode (batch/interactive) and if interactive, by your build flags.
# See "Choosing a graphics viewer" in the Application Guide for details.
# For example, with environment variable G4VIS_DEFAULT_DRIVER:
# The format is <graphics-system> [<window-size-hint>]. Set this, e.g:
# (bash) export G4VIS_DEFAULT_DRIVER=TSG
# (tcsh) setenv G4VIS_DEFAULT_DRIVER OI
# or on the command line, precede the app invocation, e.g:
# G4VIS_DEFAULT_DRIVER=Vtk ./<application-name>
# The window-size-hint can optionally be added, e.g:
# (bash) export G4VIS_DEFAULT_DRIVER="OGLSX 1000x1000-0+0"
# Other suggestions for G4VIS_DEFAULT_DRIVER (see list of registered
# graphics systems printed at the start):
# DAWNFILE: to create a .prim file suitable for viewing in DAWN.
# HepRepFile: to create a .heprep file suitable for viewing in HepRApp.
# VRML2FILE: to create a .wrl file suitable for viewing in a VRML viewer.
# "TSG_OFFSCREEN 1200x1200": to create an image file with TSG.
# See the tsg_offscreen.mac in examples/basic/B5 for more commands
# to change the file format, file name, picture size, etc.
```

# New or improved and retired vis drivers

- New in Geant4 11.0 and further developed for 11.1 and 11.2
  - Qt3D (John Allison): limited functionality but nice
  - ToolsSG (TSG) (Guy Barrand): working nicely
    - Most features of the OpenGL drivers
    - Also supports plotting
    - Full-screen driver, TOOLSSG\_OFFSCREEN—always built, default in batch mode
  - Open Inventor Qt (OIQt) (Fred Jones): Also very nice, requires users
    - Includes “bookmarking” and “navigation”
  - Vtk (Stewart Boogert, Laurie Nevay): Improved multi-featured version on the way
    - Interactive cutting and clipping
    - Export to GLTF (modern 3D object transfer protocol), interface to other packages
    - Export to web – scene can be rendered and manipulated in a webpage, “fantastic for manuals and documentation”
    - Off-screen rendering
- Retired (removed) in Geant4 11.1
  - HepRep/Wired (HepRepFile/HepRApp is retained)
  - VRML1 (VRML2 is retained)
  - The “network” drivers (those that communicate with their browser via BSD sockets)
    - VRML2 (VRMK2FILE is retained)
    - DAWN (DAWNFILE is retained)

# What about Qt6 OpenGL driver ?

- Difficult to migrate and maintain OpenGL across ALL viewers
  - Qt6 OpenGL is object oriented whether Qt5 is C code
  - Viewers have to continue working even X rendering viewers (no Qt inside)
- Lot of migration work has been done
- More work to do about OpenGLContext and Multithreading
  
- At the moment we do not provide OGLQt with Qt we asking ToolsSG to stand in for OpenGL