

Visualization Done Since Geant4.8.0

(excludes internal improvements too obscure for end users)

-

Added /vis/geometry/
for editing vis attributes of geometry volumes.
vis-V08-00-0

-

Added /vis/scene/transientsAction
as first attempt of drawing same transients to multiple visualization drivers
(not reliable in all cases - work continues)
vis-V08-00-0

-

Added /vis/modeling/trajectories/drawByOriginVolume
third of what is now three options for trajectory modeling
vis-V08-00-03

-

Added /vis/filtering/trajectories
vis-V08-00-04

-

Added 2D Text
vis-V08-00-04

-

Added /vis/list
to show current set of available drivers, models and filters
vis-V08-00-07

-

Added /vis/scene/add/eventID
vis-V08-00-07

-

Added /vis/viewer/clearTransients
needed for movies
vis-V08-01-00

-

Generating G4Atts in G4PhysicalVolumeModel.
vis-V08-01-01

-

Remove Establish/DecommissionSpecialss.
vis-V08-01-01

-

Added /vis/viewer/set/lineSegmentsPerCircle.
Can specify this in the vis attributes for specific volumes.
vis-V08-01-01

-

Added /vis/oglx/set/printEPS
Tells OpenGL X viewer that next time it updates, it should also print to file (Vector PostScript).

vis-V08-01-01

-

Added trajectory time-slicing.

Key feature for making movies.

Takes each step and breaks it down into as many vis primitives as required to allow selection in 0.1 ns steps (user selectable). Added to modeling, used at least by OpenGL.

vis-V08-01-02

-

Added /vis/ogl/set/startTime and endTime

Used for movies.

vis-V08-01-02

-

Added /vis/ogl/set/fade

Used for movies.

vis-V08-01-04

-

Added /vis/ogl/set/displayHeadTime

Used for movies.

vis-V08-01-04

-

Added /vis/viewer/set/globalLineWidthScale

vis-V08-01-05

-

Augmented /vis/scene/add/volume to include intersection of clipping volume

vis-V08-01-05

-

Added /vis/ogl/set/displayLightFront

Used for movies.

vis-V08-01-05

-

Add attribute based trajectory drawing & filtering, and attribute based hit filtering.

Provides many requested features:

Filter by momentum range.

Model colour by momentum range.

Filter based on user-supplied attributes.

etc.

vis-V08-01-06

-

Added command /vis/viewer/set/explodeFactor.

vis-V08-01-06

Visualization Public To Do List

(these items are appropriate to show to the general user community)

-

Redrawing Events on Switching to a Different Vis Driver.

Needed to allow same transients in two different vis drivers, both for use case of users who has set up two drivers in advance (e.g., please send each event to OpenGL plus DAWNFILE), or for user who scans many events in one driver (e.g. OpenGL) and then decides to send one interesting event to a different driver (e.g. DAWNFILE).
Solution will be based on changes in run management to allow keeping events.
John has lead.

-

Visualization of Field Lines (electric, magnetic, maybe even gravitational).
Initial explorations using stream line techniques, initially in Open Inventor.
Will eventually include at least minimal solution for all vis drivers (e.g., handled at base vis level rather than by driver-specific features).
Jane has lead.

-

Parallel Geometry requires enhancements to visualization.
John has lead.

-

Allow use of G4RichTrajectory without requiring user to recode.
Currently have to write user tracking action and instantiate non-default, rich trajectory.
Would be nice to instead just have command such as:
/tracking/storeTrajectory rich
Note that for case where want rich trajectory but only for specific trajectories, can apply trajectory filtering in conjunction with rich trajectory.

-

G4SmoothTrajectory is Not Smooth Enough.
Needs more auxiliary points.
Or save enough information to recompute additional steps.
Currently have to write user tracking action and instantiate non-default, smooth trajectory.
Would be nice to instead just have command such as:
/tracking/storeTrajectory smooth

-

i_mode: Remove at release 9.0.
Handled better by Trajectory Modeling Options since release 8.0.
It was officially deprecated at 8.0 including a warning message to users.
We WILL remove it at 9.0, whenever that is.
Will not break user code, but may make user code behave differently, since this parameter will be ignored.

-

Improve Boolean Operations.
Some Solids Cannot be Modeled by G4Polyhedron.
Generic Sections and Cuts would be more useful if had new boolean operation "Cut" instead of "Subtract".
Note that existing commands such as sectionPlane and cutAwayPlane either work for only some drivers or just do not give the result that most users want.
G4BREPSolid is currently visualized only as its bounding box.
See John and Evgueni's proposal, summarized Oct 05 as:

Reimplement existing functionality, including:

Improved algorithms for Boolean processing;

Caching of normals (for speed);

Offer the option of user-supplied normals (for space saving and speed);

Add a new Boolean operation – cut – that creates open polyhedra for cutaway views;

Complete the implementation of polyhedron representations of all Geant4 shapes.

-

Capture and handle ctrl-c to return to idle prompt during visualization.

End the event loop or the RayTracerX process but don't end the entire Geant4 session.

See <http://geant4-hn.slac.stanford.edu:5090/HyperNews/public/get/runmanage/17.html>

-

New Trajectory Filter: show random subset of trajectories.

To handle the issue of very large numbers of trajectories.

So, for example, I could ask to just show 1 out of 5 trajectories

-

New Trajectory Filter: filter by fraction of primary energy (rather than by absolute energy).

Requested by Steve Sekula.

-

New Trajectory Filter: filter by how many generations from primary

(eg., so me only primary, secondaries and third generation, but no further).

Requested by Steve Sekula. Vis hypernews item 341 would benefit from this.

-

New Trajectory Filter: filter by destination volume and by whether deposited energy there.

Use case described in vis thread 146 as follows:

I would like to draw only trajectories of events, that hit my sensitive detector and deposit energy there. So to say, I would like to trigger DrawTrajectory method from inside ProcessHits method of my SensitiveDetector.

-

New Trajectory Model: model by what interaction type created particle.

Somewhat similar to drawByOriginVolume, but this is drawByOriginProcess (or just drawByProcess for short).

-

New Trajectory Model: model hue by particle type and at same time model intensity by momentum.

Requested by Bruce Faddegon.

-

Asymmetric Scaling: support in more drivers. At least HepRepFile and DAWN.

-

Background Color: support in more drivers. At least HepRepFile and DAWN.

-

Window Location: support in more drivers.

HepRepFile, DAWN, OGL*Xm, OGLWIN32, Open Inventor.

-

Text: support in more drivers. Currently in DAWN and OpenGL.

Nearly done (needs attention): Open Inventor. How do you position it? 2D or 3D?
Empty implementations: HepRep, VRML1/2.

-

Smooth Shading: support in more drivers.
Is this correctly handled in Open Inventor?

-

Label Trajectories or Hits with G4Atts (for drivers that do not natively support such labeling, as only the HepRep viewers now support). Would use the new 3D Text primitives. Would require new commands to select which attributes to display.

-

Pick to Show Attributes in OpenGL:

Might make OpenGL window pickable using the same trick I used to make Tektronix emulator windows pickable back in SLD. Have to get report of where on screen user picked, then regenerate picture and see when they cross this point.

Significant effort. John may add to his "big jobs" list.

-

DAWN Rendering Web Service:

Accept that some users are not going to have DAWN on their own machine.

So one could set up a web based service. You fill out a form to tell it where to find your .prim file, you fill out some other parameters that are the equivalent to running the DAWN setup GUI on your local machine, and you tell it your email address. You then hit the submit button and walk away.

Some time later, you get an email that tells you where to pick up your completed eps file.

Could also have options to process the file through DAWNCUT or DAVID.

Could even generate more views that you had asked for after the first one (if it has spare capacity).

-

Control of Auxiliary Edges (also known as Soft Edges): support in more drivers.

Command to control this is currently implemented only in OpenGL and in DAWNGUI.

-

Filtering of Geometry According to Attributes:

This is the geometry compliment to /vis/filtering/trajectories.

Already have this as geometry "culling", but only works for culling by visibility, daughter visibility and density – plus, if mother opaque, covered daughter volumes.

Could be generalized along the lines of the trajectory filters.

HepRep browsers currently let one make such cuts on the client side.

-

Support Interoperability of the Different Vis Drivers,

i.e.. - using one driver to set camera parameters/visibility in another.

e.g., use WIRED to select view and visibility, then have G4 generate a DAWN prim file that contains the appropriate visibility and have DAWN render this with the appropriate camera parameters.

Similarly, could drive RayTracer from WIRED.

Could similarly derive vis parameters from FRED or Motif version of GL.

Point is to use a fast, rotatable viewer to drive one of the slow, high quality renderers.

Simplest solution is to have WIRED write out a G4 macro that sets camera position, etc. G4 user could then just run the macro by hand.

More elaborate solutions would involve not just setting vis parameters but also vis attributes (e.g., using one driver to edit vis attributes that are then used by another driver).

-

Allow Dynamic Loading of Vis Drivers.

Probably difficult, but worth a feasibility study.

-

Complete Immediate Mode for HepRep to WIRED or FRED.

Full implementation would require Corba. Difficult, and introduces dependency on some corba implementation.

Might be able to refine current pseudo-immediate option in which client continually looks for new file on disk so that new image is displayed almost as soon as Geant4 produces it.

-

Create HepRepFile to DAWNFILE Converter, and vice-versa.

-

Add testing of Visualization to the Release Procedure.

Extend geant4/tests/test201 to include vis commands that draw to those vis drivers that involve writing a file, such as HepRep, DAWNFILE, VRML and ASCIITree (as opposed to the immediate mode drivers).

Then just need to diff these resulting files against a reference. Include both geometry and transients.

Note that we also have something within vis called test19, but there has never been an attempt to make test19 complete. It has only been developed on a “need-to-test” basis.

-

Evaluate, Improve and Extend Visualization Examples and clean up use of visualization within all novice examples.

Move anything not very basic to extended examples.

Perhaps move vis tutor part of N03 to extended examples.

Make more examples use standard trajectory, so that they can take advantage of new features such as modeling and filtering.

-

Contribute to Geant4 Image Archive.

Include not only final images but also prim, heprep and vrmf files.

-

Prepare Tutorials for RayTracer, VRML and ASCITree.

Use structure parallel to SLAC tutorials on OpenGL, DAWN and WIRED.

-

Extend OpenGL Tutorial to discuss Motif Modes.

-

Extend DAWN Tutorial to discuss immediate mode (DAWN rather than DAWNFILE), add DAVID and DAWNCUT.

Might use the following example from Bill Lockman's mail of 4/7.

Slice of the vacuum chambers from 4490 to 4500 mm obtained by running
dawn cut 0 0 1 4500 gary.prim > gary_cut4500.prim

dawncut 0 0 -1 -4490 gary_cut4500.prim > gary_cut4490_4500.prim
dawn gary_cut4490_4500.prim

I've enclosed the gary_4490_4500.eps file, obtained by azimuth=polar=0,
magnification=25, selecting viewing mode=line, and coordinate axes.

-

Extend Vis Tutorials to Show More Commands:

such as culling, sectionPlane and the detailed use of /vis/viewer/add/volume
(to limit number of volumes drawn, specifying by name and/or by depth and specifying
clipping volume).

-

Make a Tutorial about VisUserAction.

G4LogoVisAction is a good illustration. Needs a mention in the documentation.

-

HepRep1: Need to add initial scale and view angle hints to the HepRep standard.

Then make HepRep1 driver produce this hint.

Users probably want their detector to fit, but not necessarily all trajectory points to fit.

-

Simplify/clarify "Configure -build" questions about Vis Drivers.

At least make more clear that can have things like DAWNFILE without having to say yes
to immediate mode DAWN.

Might make the vis part of the dialog begin by saying:

"You will automatically have available. xxx, yyy, zzz..

Do you want optional additional visualization drivers?"

-

Remove need to set WIN32_SESSION to "Configure -build" with OpenGL on Windows.

Need to do this at present to force the G4INTY variable.

From ./Configure -build, one has to do this by replying yes to the question about:

G4UI_BUILD_WIN32_SESSION

G4UI_USE_WIN32

Why should this be necessary?

-

Why do we have different OpenGL variables for Windows and non-Windows.

System already knows if it is windows or not based on the G4SYSTEM variable.

Even if we do need the Windows versus non-Windows G4 vis variables behind the
scenes, why do we need to expose this to the user? In particular, why have them use
different OpenGL /vis/open commands on the different operating systems?

Should just always be /vis/open OGLxy, with no Win32 in there.

-

Provide guidance to users on how to make OpenGL windows refresh on uncovering.

This is not strictly a Geant4 Visualization issue, but it is a ubiquitous issue for our users.

-

Configure -build on Mac with OpenGL gives annoying, non-fatal warnings, such as:

Creating/replacing object files in /Applications/geant4/lib/Darwin-g++/libG4OpenGL.a

...

ar: creating archive /Applications/geant4/lib/Darwin-g++/libG4OpenGL.a

ranlib: file: /Applications/geant4/lib/Darwin-g++/libG4OpenGL.a(G4OpenGLImmediateWin32.o) has no symbols
ranlib: file: /Applications/geant4/lib/Darwin-g++/libG4OpenGL.a(G4OpenGLImmediateWin32Viewer.o) has no symbols
ranlib: file: /Applications/geant4/lib/Darwin-g++/libG4OpenGL.a(G4OpenGLImmediateXm.o) has no symbol

-
Configure -build on Windows with OpenGL gives annoying, non-fatal warnings, such as:
Making dependency for file src/G4OpenGLStoredSceneHandler.cc ...
src/G4OpenGLStoredSceneHandler.cc:144:31: macro "max" requires 2 arguments, but only 1 given
src/G4OpenGLStoredSceneHandler.cc:144:40: macro "max" requires 2 arguments, but only 1 given

-
Implement /vis/scene/add/title [<title="Geant4">] [<size>] [<x>] [<y>] (2D)

-
Implement /vis/scene/add/date [<size>] [<x>] [<y>] (2D)

-
Implement /vis/scene/add/logo2D

-
Implement /vis/scene/add/text2D

-
Once Scorers have Generic Hits, make sure Vis Handles them Well

Visualization Private To Do List

(too obscure or too poorly thought out to show to the general user community):

-
Need a Realistic Complex Geometry for Benchmarking.
Problem is most experiments embed geometry construction in their own framework.
CMS geometry is available to us in GDML, might do the trick.

-
Add New G4Att for Trajectories: OriginVolume

-
Make a way to show DICOM data in WIRED. Maybe can use cuts by position.

-
Support Qt as Wrapper for Visualization.

Would have to write new session, G4UIQtSession, would have subwindows, one of which could be filled with a graphics window.

Would have to have OGLIQt, OGLSQt.

-
Review RayTracer Documentation.

See John Allison's mail of 23 Jan 06. Is guidance at least complete? John committed some improvements around July 06 to
documents/UserDoc/UsersGuides/ForApplicationDeveloper/html/Visualization/visdrivers.html.

- Solve LGPL Freehep License Issue in HepRep2 driver.
- Deprecate SetVisAttributes (const G4VisAttributes& VA).
See John's mail of 13 Sept 2006. Topic came up as "memory leak in G4Visible".
- Extend Vis Modeling Commands to TrajectoryPoints
- G4HepRepFileSceneHandler can have easier time Fixing Hierarchy for Culled Layers since we now have the unculled depth stored in the PV so that `drawnMotherDepth = ri->GetNonCulledDepth()`
John indicated that he had a fix waiting to commit. So maybe it is now done.
- John's comments about Motif for Mac:
Another point is that Darwin-g++.gmk was written by Guy some time ago and he installed motif in /sw, but the standard place - at least for the now available OpenMotif, is /usr/X11R6. But changing it might cause havoc in the Mac community...Mmmm! No, I doubt it. My guess is that there is not more than a handful of Mac Motif users. How about polling the user community to see how many, and where they have installed Motif, and what version. How about asking Guy his views on this?
This is settled, I believe. Guy argued to keep the flags pointing to /sw.
- Add AddCompounds for Axes and Ruler.
Would allow HepRep driver to treat these more intelligently.
An alternative is for the G4AxesModel and G4ScaleModel to attach a G4AttValue that signifies what the primitives are representing - a string "axes", "scale", or some such. Could have GetAttDefs and CreateAttValues in the G4VMModel base class. Require every model to announce itself. Perhaps trajectories, hits and digis too should announce themselves in this way.
- Trajectory Step-Slicing.
Somewhat analogous to trajectory time-slicing, somewhat analogous to RichTrajectory. Each vis primitive - line, circle, square, etc. - should have all the appropriate attributes. At present the lines get the trajectory attributes and the circles get the trajectory point attributes. But the line has just as much claim on the particle time, for example, as the circle - in fact more so, for the pre and post times correspond to the start and end of the step, whereas the circle is placed at the end of the step.
What we do, in effect, is take each step (represented by a trajectory point) and break it in several vis primitives according to the users choices - a line (or several lines if auxiliary points are present) and a marker (or several markers of various types if auxiliary points are present). So I think each of those should have the trajectory attributes (so that one can select all of them on particle ID, for example) *and* the trajectory point attributes for that trajectory point (so that one can select all of them on particle time, for example).
- Could add Line Style to vis attributes (solid, dotted, dot-dash, etc.).
Not currently used by any driver. Maybe not important given the ubiquity of color.

-
Line Width attribute: make more drivers correctly use it. What is the current situation?

-
SetDaughtersInvisible: Unresolved issues about what behavior users might expect.
Excerpt from last email on this between Joseph and John was 6 Feb 2006:
This attribute has been around for years. The idea, as far as I can remember, was to have an efficient way of simplifying a drawn tree if the detailed sub-volumes were not required to be seen. If true, the descendants are not even inspected; the recursive descent is curtailed. However, I've never heard of it being used. I have just done a grep of all examples, and it is not used. Nevertheless, let's see if it can stay.
By the nature of its implementation, it is a flag in a mother that affects all descendants. It is not the same as recursively applying the invisible attribute. If the flag is toggled, all the descendants become invisible/visible, or rather, all invisible/descendants vis attributes apply. An of course, only applies at all if global culling is on.
Also, by its nature, it doesn't make sense (or does it?) to propagate it recursively to descendants, so although the command structure allows for a depth parameter (to make it similar to the other commands), it is suppressed as I have coded it in this case.

-
Try DICOM and VRML (and iv) imaging suite recommended by Andreia Trindade:
Amira: <http://www.amiravis.com>
Amide: <http://amide.sourceforge.net/index.html>
Also explore gMocren.

-
What Geant4 Vis Drivers are Provided in Mulassis?
Suspect that only VRML is currently supported. Should they provide others?

-
Immediate modes in general, and for the interoperability issue in particular, need clear statements of what platforms to test and support.

-
Remove "BookKeeping" from attribute setting code. This is not legal HepRep.
Should just be "Physics" (other legal options are Draw, Association and PickAction).
G4AttCheck.cc
G4RichTrajectory.cc
G4RichTrajectoryPoint.cc
G4SmoothTrajectory.cc
G4Trajectory.cc
Currently, replace this on the fly in HepRepFile, but it might be present in HepRepXML.

-
OpenGL: If you start a run with more than one event, and the you have asked for the scene to be refreshed at the end of each event, the interactive viewer OGL*Xm goes interactive mode at the end of each event, as expected. If you ask then for something that requires a re-computation of transients, for any other than the last event, the run manager silently refuses to recompute the transients because it's already inside a run. You can't have a run within a run. The last events's OK, because the vis manager postpones the ShowView until the end of run.
Non-critical issue. Review when event refreshing settles down.

-

DAWN: Check support for gcc 3.4.3

-

DAWN: John's note in vis hypernews of 13 June 26, 2006 says:

DAWN needs wish. wish is a windows supplement to tcl/tk.

Is this part of cygwin dev? Otherwise, why haven't I had to tell people to explicitly load it? John says he thinks wish is part of any decent Unix system.

-

HepRep1: Take advantage of following new features:

3rd July 2006 John Allison (greps-V08-01-00)

G4AttCheck.cc:

- o Added G4bool to the list of allowed G4AttValue value types.
- o Added Density to the list of units categories.

G4AttDefStore: Added GetName.

-

HepRep1: remove extraneous level from detector hierarchy:

Detector Geometry[0] has subtype, Detector Geometry[0].

-

Find cleaner solution to Avoiding Drawing Geometry More than Once to the same file.

Current solution uses Rewind (file close followed by file-reopen).

Joseph doesn't like that current solution. John thinks it's not so bad.

-

HepRep1:

Find a way to make

/vis/scene/create

/vis/open HepRepFile

/vis/viewer/add/volume

/vis/viewer/flush

not make double geometry.

When I do /vis/scene/add/volume, geometry is sent out to my heprep file.

When I do /vis/viewer/flush, the same geometry is sent out again.

I wish I could make /flush know to only send out the geometry if there is something new, but I don't see how.

I'm wondering again whether, for HepRepFile, it is OK to just move the

```
NeedKernelVisit (); // Always need to visit G4 kernel.
```

```
ProcessView ();
```

From DrawView to ShowView.

Maybe do move the action to ShowView.

Fails with my test3.mac.

HepRepFile writing to G4Data0.heprep

G4VisManager: Using G4TrajectoryDrawByCharge as default trajectory model.

See commands in /vis/modeling/trajectories/ for other options.

G4VisManager::GetInstance: VisManager not yet instantiated!

Study the rewind option from John.

-

HepRep1: Any More Solids that can be More Efficiently Modeled:

rather than deconstructed into polygons?

-

HepRep1: use GetDesc rather than GetName once WIRED3 can Handle Names with Spaces in them.

hepRepXMLWriter->addAttValue(iAttDef->second.GetDesc(), iAttVal->GetValue());
Want to see the description in the label or cuts window because hard to remember what the cryptic name means (e.g., TED for Total Energy Deposited).

-

HepRep1: how does it respond to /vis/viewer/set/lineSegmentsPerCircle.

Does WIRED3 look for this information in any particular attribute? If not, add to HepRep standard.

-

HepRep1: take another look at Handling of Culled Volumes.

Future improvements might assign correct names to the missing parent volumes. This might then make more correct replacement hierarchy, where currently all culled situations go to their own top level culled volume.

-

HepRep1: Handle GetMarkerSize with Value in World System.

Currently just handles cases where the value is delivered in the screen system.

-

HepRep1: improve visibility for G4HepRepFile Environment Variables:
or replace them with commands.

-

HepRep1 and 2: Commands /vis/heprep only work for HepRepXML.

Either use them in HepRepFile, or change the commands, or at least fix the guidance.

For now, have at least corrected the guidance that is in the AppUserGuide.

-

HepRep2: IMAG Problem and general issue of default values.

Gives the following error for each event:

ERROR: No HepRepDefaults, trying to get value for: IMag

Doesn't seem like this driver has a sensible design for user-supplied attributes.

-

HepRep2: allow geometry for complex detectors to be written to several separate HepRep files so that user can load just the parts that are wanted (each detector component may even have two versions, one minimal, one very complex).

-

HepRep2: seems to have neutrals with charge "0" rather than "0.".

-

HepRep1 and 2: solve confusing names of drivers.

Make them HepRep1File and HepRep2File.

-

Review my "HepRep3" mail of 19 Dec 2003 which responded to some of John's ideas about G4 HepRep.

-

OpenGL: Support haloing.

-

OpenGL: Support anti-aliasing.

-

Begin/EndPrimitives2D: handle in more drivers.

Currently just done for OpenGL and HepRepFile.

-

HepRepFile: Make Axes Lines have Attribute that Tells Length of Scale

-

G4TesselatedSolid: Are Auxiliary Edges Correctly Handled?

-

Remove remaining vis managers from geant4/tests and any other examples

-

Remove References to XML Tree in App Dev Guide 8.10.

This driver never actually existed.