

Vis and UI in examples

John Allison

Geant4 Associates International Ltd
and
The University of Manchester

History

- The first examples were written almost 20 years ago. Their design has been copied down the years.
- `main()` uses flags `G4UI_USE` and `G4VIS_USE` (for reasons explained on the next slide). They are very rarely used (they are always set) and, for reasons to do with simplifying the build system, we wish to eradicate them.
- The introduction of Graphical User Interfaces places some new requirements on the design of `main()`.

G4UI_USE and G4VIS_USE

- Geant4 was—and still is—capable of being built without a vis system and without a user interface. In the past a significant saving in memory usage could be made this way, but this is no longer an issue. This is achieved by requiring all Draw messages to be sent to the low-level G4VVisManager interface in graphics_reps. This will still be the case.
- In early days we highlighted this capability by introducing environment variables G4VIS_USE and G4UI_USE, which are translated into the C-pre-processor flags of the same name, which had to be set to include appropriate code. However these are always set in the standard environment. We would like to eliminate them to simplify the build system and clean up the code to make it easier for users to read.
- Thus we recommend that the vis manager and the user interface always be instantiated, even in batch mode. If they are not used they simply occupy an insignificant amount of memory for code (along with all the unused physics models!).
- Note: You may use the vis manager in batch mode with /vis/ commands in the batch macro, for example to write a file for off-line browsing.
- Note: Although writing your own drawing code is still allowed (with some restrictions in multithreaded mode) it is not needed so much nowadays. The drawing requirements of most users can be satisfied by vis commands, such as /vis/modeling/trajectories/... If nevertheless you need to do your own drawing, it is still allowed but it is mandatory that Draw messages are written to the G4VVisManager interface and protected:
 - `G4VVisManager* pVVisMan = G4VVisManager::GetConcreteInstance();`
 - `If (pVVisMan) {`
 - `pVVisMan->Draw(...`
 - *It is recommended that Draw messages are encapsulated in the Draw methods of G4VTrajectory or G4VHit or a “vis user action”. This way the vis manager has control and can refresh or re-write as required. If these do not meet your requirements you may still write Draw messages but if issued inside the event loop they will currently be ignored in multithreaded mode (this restriction could be lifted if there is a strong use case).*

Graphical User Interfaces

- Another issue has arisen because of the introduction and common usage of graphical user interfaces, such as G4UIQt.
- These capture G4cout, so it should be instantiated at the very beginning of main(), otherwise some output will go to std::cout/cerr. Many examples instantiate the UI too late.
 - An exception might be that you wish to validate the environment and/or data and exit the main program in the case of failure. Interface instantiation should come after this so that any output reporting on the reason for failure goes to std::cout/cerr. Otherwise will be lost.
 - Another exception might be that you are simply not interested in early output.

	Old (TestEm1.cc)	New (exampleB1.cc)
#include statements	<pre>#ifndef G4VIS_USE #include "G4VisExecutive.hh" #endif #ifndef G4UI_USE #include "G4UIExecutive.hh" #endif</pre>	<pre>#include "G4VisExecutive.hh" #include "G4UIExecutive.hh"</pre>
In interactive mode, G4UIExecutive is instantiated straight away so that for graphical user interfaces all G4cout output is captured from the start. ui will be used as a flag for interactive mode.	<pre>int main(int argc, char** argv) { ...</pre>	<pre>int main(int argc, char** argv) { G4UIExecutive* ui = 0; if (argc == 1) { ui = new G4UIExecutive(argc, argv); } ...</pre>
The logic here is maintained. Note: it's usual to invoke vis.mac or init_vis.mac in interactive mode before starting the session.	<pre>G4UImanager* UI = G4UImanager::GetUIpointer(); if (argc!=1) //..batch mode { G4String command = "/control/execute "; G4String fileName = argv[1]; UI->ApplyCommand(command+fileName); } else // interactive mode { #ifndef G4VIS_USE G4VisManager* visManager = new G4VisExecutive; visManager->Initialize(); #endif #ifndef G4UI_USE G4UIExecutive * ui = new G4UIExecutive(argc, argv); ui->SessionStart(); delete ui; #endif #endif #ifdef G4VIS_USE delete visManager; #endif }</pre>	<pre>G4VisManager* visManager = new G4VisExecutive; visManager->Initialize(); G4UImanager* UImanager = G4UImanager::GetUIpointer(); if (! ui) { // batch mode G4String command = "/control/execute "; G4String fileName = argv[1]; UImanager->ApplyCommand(command+fileName); } else { // interactive mode UImanager->ApplyCommand("/control/execute init_vis.mac"); ui->SessionStart(); delete ui; } delete visManager;</pre>

Summary

- Basic examples, introduced 4 or 5 years ago, have always had this form.
- From 10.5.beta, all extended examples have this form.
- Next: advanced examples, tests—and maybe category tests.