

Updates on Vis and UI's

Laurent Garnier OSUR / INSU / CNRS

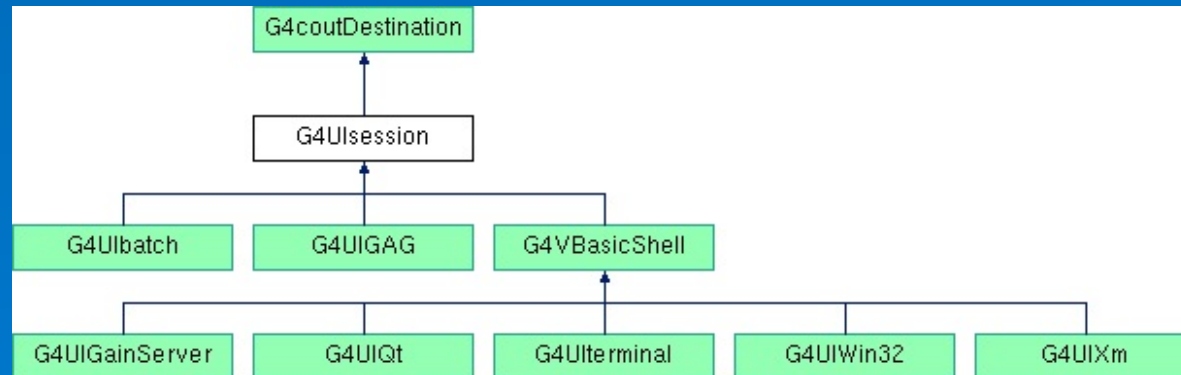
Geant4 Collaboration Meeting – Chicago – 28 September 2015

Updates on Vis and UI's since 10.1 beta

- 1. Quick overview of UI's
- 2. What's new in UI
- 3. Updates on Vis
 - Multithreading in Visualization
 - C++11
 - Re-implemented graphics system fallback strategy
 - Fix a problem when zooming on small volumes

Quick overview of UI's

- What do we have :



- **G4UIQt**, the most up to date viewer, base on Qt framework
- **G4UITerminal**, the default user interface of Geant4,
- **G4UIXm**, **G4UIWin32**, variations of the upper terminal by using a Motif or Windows widget
- GAG, a fully Graphical User Interface and its extension **G4UIGainServer** of the client/server type.

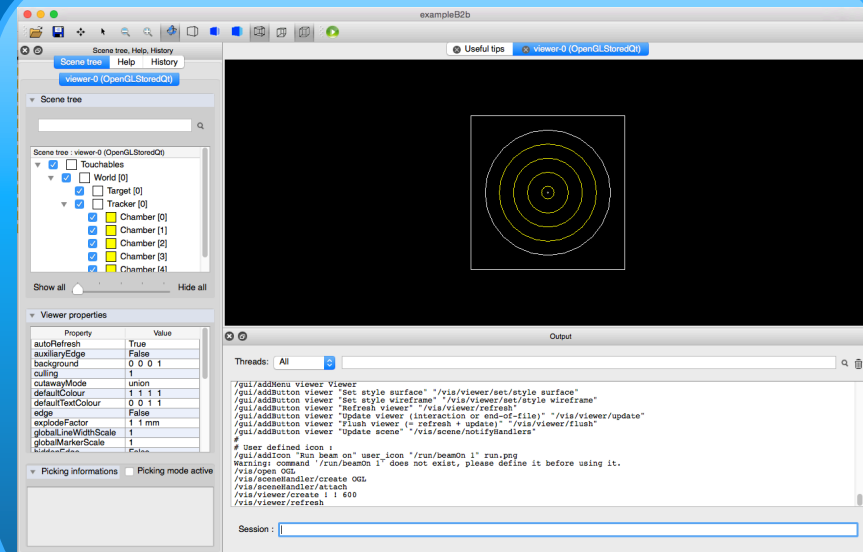
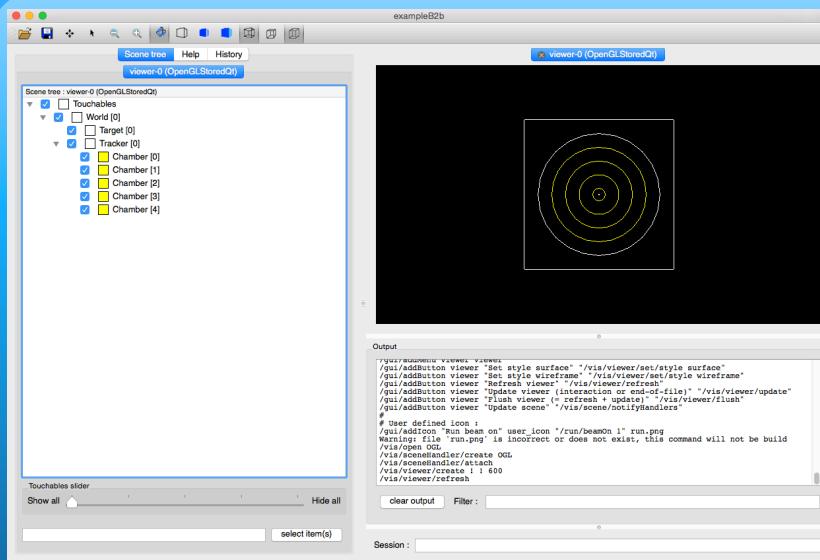
What's new in User Interfaces ?

- No major new features developed
- Bugs fixes
- G4UIQt :
 - add methods to access user interface components

Updates on G4UIQt

Geant4.10.0.p5 with Qt4.8

Geant4.10.2 with Qt5.5

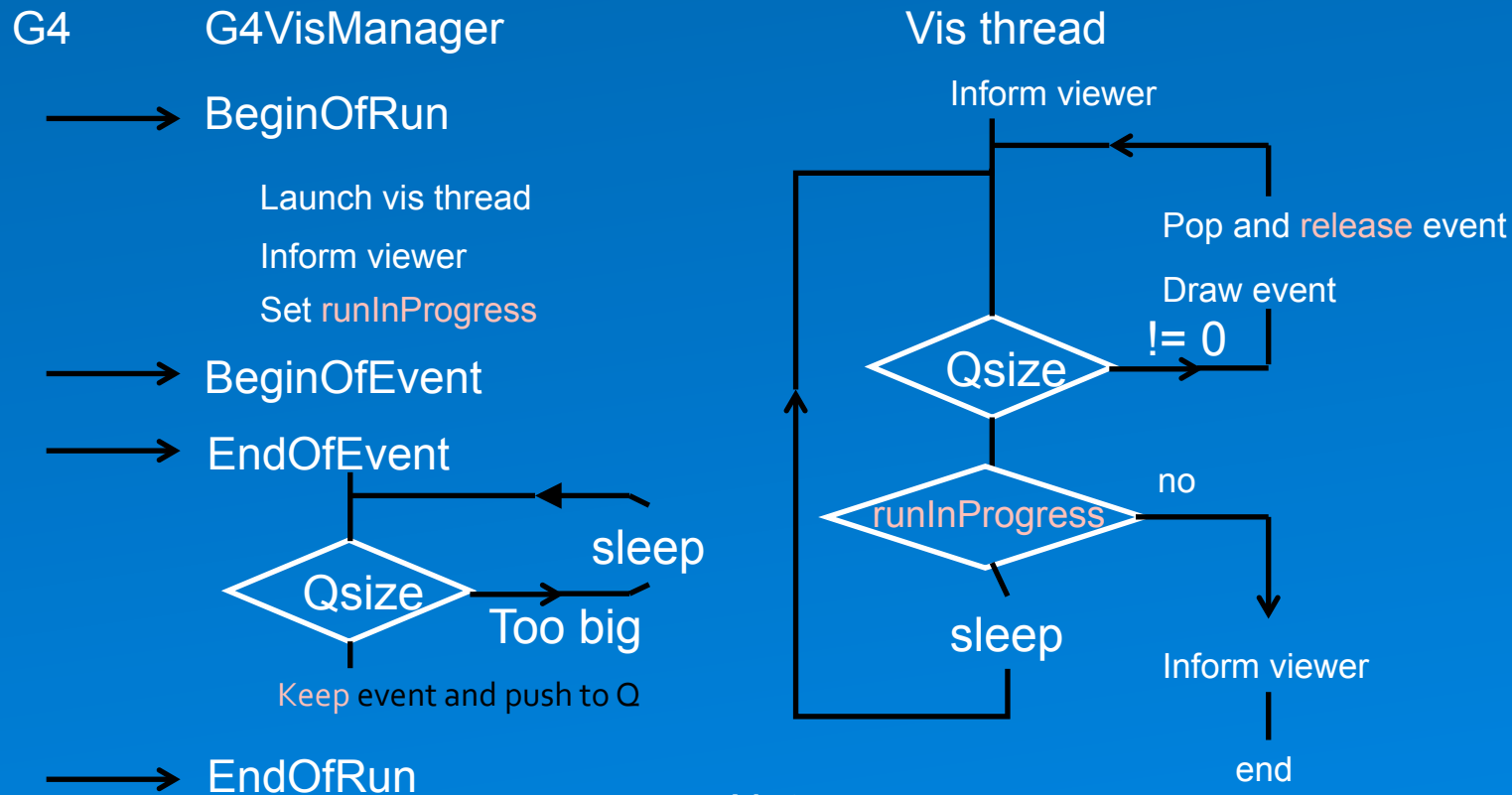


Updates on Vis : Multithread

(Slides from John Allison)

- **Aim:**
visualisation behaviour in multithreading mode should be as close as possible to sequential mode
- **In 10.0 and 10.1**
Events (default 100) are saved and displayed at end of run
Clearly different to sequential, but good enough for checking
- **From 10.2**
Events are queued and drawn by a separate vis thread during the run
To the user, multithreaded behaves same as sequential
As users adopt multithreading they require full vis features

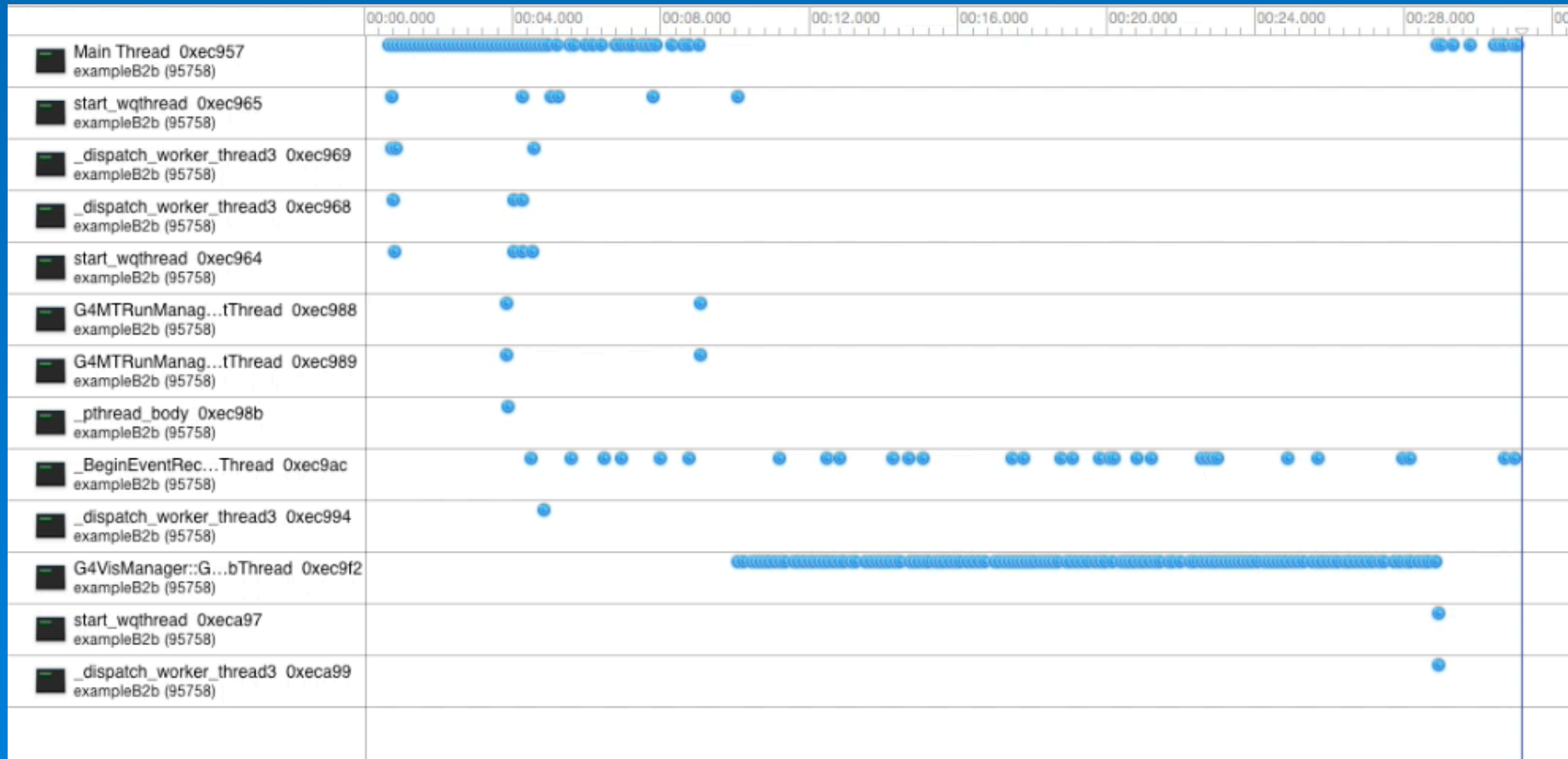
Vis flow diagram - multithreaded mode



Notes:

- Queue is `std::deque<const G4Event*>`
- Appropriate locks are used to set and reset `runInProgress`
- “**Keep**” and “**release**” refer to messages to the run manager (new feature)

Vis thread



Updates on Vis

- C++11
- Re-implemented graphics system fallback strategy
- Fix a problem when zooming on small volumes

Question ?
