New Developments In Analysis

I. Hrivnacova, G. Barrand,
IJCLab Orsay (CNRS/IN2P3)

25th Geant4 Collaboration Meeting,
25 September 2020

Outline

- Updates in g4tools
- Updates in g4analysis in 10.6
  - Improved ntuple merging
  - Utility functions/commands for batch plotting
- New developments for 10.7
  - New “Generic” analysis manager handling multiple output files and output types
- Plans for 2021

g4tools since 2019

Geant4 collaboration meeting Sep 2020

Mainly fixes

- ROOT format : restore a "per event/row view" for the ntuple column-wise in MT and MPI.
- offscreen plotting :
  - fixes to avoid labels overlap when having a grid of plotters.
  - fixes to handle automatic min/max axes values in case of log scale.
- Various fixes to make Coverity happy.
- Fixes to make clang10 and gcc-9.2 happy.

Updates in g4analysis in 10.6

Ntuple Merging

- Ntuple merging is available for ROOT output type since 10.3
- The `SetNtupleMerging()` function , defined in 10.5 in G4RootAnalysisManager with 4 arguments, was split into four functions and moved to G4VAnalysisManager
  - No need to comment out this call (eg. in basic examples) when switching the output type
  - A warning is issued when called with output types other than ROOT
-

```
// MT/MPI
virtual void SetNtupleMerging(G4bool mergeNtuples,
                              G4int nofReducedNtupleFiles = 0);
virtual void SetNtupleRowWise(G4bool rowWise, G4bool rowMode = true);
virtual void SetBasketSize(unsigned int basketSize);
virtual void SetBasketEntries(unsigned int basketEntries);
```

Ntuple Merging Modes

- New in g4tools: restore a "per event/row view" for the ntuple column-wise in MT and MPI
  - This new mode became the default in 10.6 with keeping two previous modes optional
- A new function introduced to define the merging mode:

```
void SetNtupleRowWise(
        G4bool rowWise, G4bool rowMode = true);
```

- With the default, *column-wise with preserving rows mode:*

```
rowWise = false, rowMode=true
```

  - The column-wise mode enhanced with preserving the ntuple rows
  - It requires larger memory size than the other two modes depending on the users applications.

Ntuple Merging Modes (2)

The optional modes (existing already in the previous Geant4 versions):

- *column-wise not preserving rows*

  `rowWise = false, rowMode=false`

  - The fastest option, but without preserving the ntuple rows after merging.

- *row-wise*

  `rowWise = true`, `rowMode` value is not used

  - Introduced in 10.4 and became a default in 10.5. In this mode, columns are defined as leaves of a single TBranch attached to each ntuple per worker.
  - This approach has an inconvenience that as column-wise is used in sequential mode and row-wise is used in parallel, the user will have different data schema (different organizations of TBranches and TLeaves) in files, which may complicate his life when reading back his data.

Other Enhancements

- New functions for histogram & profiles attributes for plotting:

```
// Setters for histogram & profiles attributes for plotting
G4bool SetH1Title(G4int id, const G4String& title);
G4bool SetH1XAxisTitle(G4int id, const G4String& title);
G4bool SetH1YAxisTitle(G4int id, const G4String& title);
G4bool SetH1XAxisIsLog(G4int id, G4bool isLog);
G4bool SetH1YAxisIsLog(G4int id, G4bool isLog);
//
```

New in 10.6

  - Activate axis log scale for plotting
  - Available for the axis for all histogram/profile types

- Corresponding UI commands:

```
/analysis/h1/setTitle id title  # Set title for the 1D histogram of #id
/analysis/h1/setXaxis id title  # Set x-axis title for the 1D histogram
/analysis/h1/setYaxis id title  # Set y-axis title for the 1D histogram
/analysis/h1/setXaxisLog true|false  # Activate x-axis log scale for plotting
/analysis/h1/setYaxisLog true|false  # Activate y-axis log scale for plotting
```

New in 10.6

New Developments
For 10.7

In development branch:
https://gitlab.cern.ch/ihrivnac/geant4-dev/-/tree/analysis-redesign

Using Multiple Files

- Users can choose the output file per object via new analysis manager functions:

```
void  SetH1FileName(G4int id, const G4String& fileName);
    //… etc for H2, H3, P1, P2
void  SetNtupleFileName(G4int id, const G4String& fileName);
```

  - The file name should be provided with the extension
  - Multiple output types can be used for histograms and profiles, only one output type for ntuples

- The corresponding UI commands will be also available

```
/analysis/h1/setFileName id  myFile1.ext
/analysis/h1/setFileNameToAll  myFile2.ext
  # etc. for h2, h3, p1, p2
  # ext can be csv, hdf5, root, xml
```

Design Changes

- Handling more files by analysis manager required a deeper redesign of the manager classes
- Separation of the objects output from the analysis managers
  - New base classes introduced in the ‘management’ sub-category: G4VTHnFileManager<HT>, G4VNtupleFileManager
  - Their specialization in the output specific categories: G4RootHnFileManager<HT>, G4RootNtupleFileManager. etc
    - G4RootNtupleFileManager also takes care of management of all helper classes for MT merging
- Common implementation of ntuple booking in G4NtupleBookingManager into `tools::ntuple_booking` objects independent from the output type
- New factory classes in the ‘factory’ sub-category: G4GenericAnalysisManager, G4GenericFileManager

New ‘factory’ Classes

- G4GenericAnalysisManager takes the role of the top analysis manager class
- Derived from G4ToolsAnalysisManager, which provides handling of histograms/profiles except for file output
- Uses new G4NtupleBookingManager providing booking of ntuples independently from the output type
- The output type dependent managers for file and ntuple management are created at `OpenFile()` call by G4GenericFileManger
  - The file managers can be more than one, they are then handled by the generic file manager
  - Only one ntuple file manager, it is then passed to the generic analysis manager
    - Only one output type for ntuples will be supported

Instance Methods

```
#include "G4GenericAnalysisManager.hh"

// Create analysis manager
auto analysisManager = G4GenericAnalysisManager::Instance();
```

NEW (10.7)

```
//#include “g4csv.hh”
//#include “g4xml.hh”
#include “g4root.hh”
```

#include "B4Analysis.hh"

```
// Create analysis manager
// The choice of analysis technology is done via selection
// of a namespace in B4Analysis.hh
auto analysisManager = G4AnalysisManager::Instance();
G4cout << "Using " << analysisManager->GetType() << G4endl;
```

ALL VERSIONS

Instance Methods (2)

```
#include “g4analysis.hh”

// Create the analysis manager using a new factory method.
// The choice of analysis technology is done via the function
// argument.
auto analysisManager = G4Analysis::ManagerInstance(“root”);
G4cout << "Using " << analysisManager->GetType() << G4endl;
```

SINCE 10.6

- All three methods will be supported in 10.7
- The output specific analysis manager classes and the headers (“g4csv.hh, etc.) may be dropped next year in favor of the new generic manager

Writing to (an) Extra File(s)

- In addition to the previous mode, where each object has associated its file, an extra Write call can be performed to write (a) selected object(s) to another file:

```
G4bool WriteH1(
          const G4String& fileName, G4int id = -1);
```

  - If id = -1 all histograms will be written
  - Also for H2, H3, P1, P2, not for ntuples
- The file will be closed at this function, the histograms will not be reset, in MT this will trigger merging from worker to master
- Still experimental

Other Developments & Plan

- Font License issue
  - Bug report #2267: *Enabling GEANT4_USE_FREETYPE leads to the installation of non-free fonts*
  - Solution: replacement of currently used fonts with OpenSource; we have good candidates that we can read with the existing g4tools code
- Additional flexibility in resetting/deleting histograms
  - Resetting selected histograms can be added for 10.7
  - Deleting selecting histograms will be moved for 2021
- Review support for writing same histogram/profile on file several times
  - Will be moved for 2021
- Attempt to integrate g4tools/plotting in the G4/vis and UI system
  - Will be moved for 2021