Updates to Analysis Module

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

27th Geant4 Collaboration Meeting, Rennes,
29 September 2022

Outline

- Updates in g4tools
- Updates in analysis
  - Migration to Generic Analysis Manager in 11.0
  - Developments for 11.1
- 2022 Work plan items

g4tools @ Rennes-2022

tools / toolx

- Put in the toolx directory and under the toolx namespace all what is related to external packages.
- Then for what is related to G4/analysis, code related to expat, zlib, mpi, hdf5, freetype.
- And for what is related to G4/visualization, code related to OpenGL, X11, Xt, Windows, Qt.
- All what is on the std C/C++ libs (and some standard OS functions) remain under the tools directory and tools namespace.
- It should help the readability of the code.

As a reminder...

- g4tools is an automatic extraction of some code found in the softinex/inlib,exlib and namespaced with tools/ toolx for an embedding in Geant4.
- Pure header code. Highly portable (including iOS, Android and now WebAssembly). Easily embeddable (no “config.h” or specific build tool in the way).
- Strongly OO. No implicit management. Layered.
- Thread safe (no writable statics).
- See http://gbarrand.github.io

New Developments in 11.0

Migration to Generic Analysis Manager in 11.0

- G4GenericAnalysisManager – first introduced in 10.7 and became the default in 11.0
  - It allows to select the output file type at run time, supports multiple files, including multiple output types for histograms and profiles
  - Still limitation of one output type for n-tuple management
- Users can still use the output specific manager, eg. G4RootAnalysisManager etc.
  - They need to include it directly
  - No deprecation warning is issued

- A new header file G4AnalysisManager.hh provided in 11.0 and used in almost all examples

```
#ifndef G4AnalysisManager_h
#define G4AnalysisManager_h

#include "G4GenericAnalysisManager.hh"

using G4AnalysisManager = G4GenericAnalysisManager;

#endif
```

- The output specific headers `g4csv.hh, g4hdf5.hh, g4root.hh, g4xml.hh and g4analysis.hh were removed`

Instance Methods in 11.0 - 1

```
#include "G4AnalysisManager.hh"

// Create/Get analysis manager
auto analysisManager = G4AnalysisManager::Instance();

// Open an output file
G4String fileName = "B4.root";
// G4String fileName = "B4.csv";
// G4String fileName = "B4.hdf5";
// G4String fileName = "B4.xml";
analysisManager->OpenFile(fileName);
G4cout << "Using " << analysisManager->GetType() << G4endl;
```

NEW in 11.0

G4AnalysisManager = G4GenericAnalysisManager

Implicit Using defined in G4AnalysisManger.hh

Output file name is provided with the extension

Instance Methods in 11.0 - 2

```
#include "G4AnalysisManager.hh"

// Create/Get analysis manager
auto analysisManager = G4AnalysisManager::Instance();
analysisManager->SetDefaultFileType("root");
// analysisManager->SetDefaultFileType("csv");
// analysisManager->SetDefaultFileType("hdf5");
// analysisManager->SetDefaultFileType("xml");
G4cout << "Using " << analysisManager->GetType() << G4endl;

// Open an output file
G4String fileName = "B4";
analysisManager->OpenFile(fileName);
```

NEW in 11.0

G4AnalysisManager =
G4GenericAnalysisManager

Implicit Using defined
in G4AnalysisManger.hh

File extension can be
omitted if default file type
is set explicitly

Instance Methods in 11.0 - 3

```
#include "G4RootAnalysisManager.hh"

// Create/Get analysis manager
auto analysisManager = G4RootAnalysisManager::Instance();

// Open an output file
G4String fileName = "B4";
analysisManager->OpenFile(fileName);
```

Explicit include of the output specific managers is also possible

New Developments for 11.1

Design Changes - 1

- Common implementation of **histogram handling** for all histogram/profiles type
  - 15 classes were reduced to 3
  - 1 interface (G4VTBaseHnManager<DIM>), 1 implementation (G4THnToolsManager<DIM,HT>) and 1 messenger (G4THnMessenger)
  - Instead of 5 interfaces, 5 implementations, 5 messengers for each histogram/profile type (H1, H2, H3, P1, P2)
    - We we still need the explicit specialization of `Create, Set, Fill` and `WriteAscii` methods (defined in G4TH*|P*Manager - 5 .hh files and 5 .cc)
    - However the code size of template specializations is ~4x smaller than the original managers

Design Changes - 2

- Refactored the analysis manager methods for **file handling** from the output specific managers in the common implementation in G4ToolsAnalysisManger
  - In 11.0 – the functions `OpenFile, Write` and `CloseFile` are implemented both in G4GenericAnalysisManager and the specific managers
- This guaranties identical behavior with the generic analysis manager
- Code simplification:
  - No “execution” methods in specific managers, only instantiation of helper managers and n-tuple access methods

Object Cycles

- Support for **writing the same histogram/profile on file several times**
  - Requested at the G4 User forum
  - MR in preparation
- If the same object is written in a file more than once, it gets automatically attributed **the cycle number**
  - Root IO supports the cycles naturally
    - When eg. "`myhisto`" is written more times, we can see in the browser `myhisto;1 myhisto;2` etc.
  - For the other output types, the cycle number is appended after the object name:
    - Eg. `myhisto.csv, myhisto_v2.csv`, etc.

Object Cycles - 2

- The function `Write()` can be now called more than once before `CloseFile()`
- This leads to the need to provide UI commands for `"open/close"` file and `"reset, clear"` functions

An example of a run macro when the object cycles are supported

```
/run/initialize
/analysis/openFile B5.root
#
/run/beamOn 30
/analysis/write
/analysis/reset
#
/run/beamOn 30
/analysis/write
/analysis/reset
#
/analysis/closeFile
```

Other Developments

- Added the "`getVector`" command for visualization (in addition to "`get`" already available in 11.0):
  - `/analysis/xy/getVector,  xy = h1, h2, h3, p1, p2, ntuple`
- Added G4AnalysisManger "`List`" functions

```
// List objects
G4bool ListH1(G4bool onlyIfActive = true) const;
G4bool ListH2(G4bool onlyIfActive = true) const;
G4bool ListH3(G4bool onlyIfActive = true) const;
G4bool ListP1(G4bool onlyIfActive = true) const;
G4bool ListP2(G4bool onlyIfActive = true) const;
G4bool ListNtuple(G4bool onlyIfActive = true) const;
G4bool List(G4bool onlyIfActive = true) const;
```

- and corresponding UI commands:
  - `/analysis/xy/list [onlyIfActive],  xy = h1, h2, h3, p1, p2, ntuple`
- Clang-tidy – applied all checks recommended in CODING_GUIDELINES.rst
- Coverity fixes

2022 Work Plan Items

- Implement features for interactive plotting (visualization) – DONE
- Support for multiple output types for n-tuples – (1)/(2)
  - Support for multiple output files is available since 10.7
  - As there is no request from users for this feature, we will drop this item from the WP; it can be added in future if needed
- Addition of flexibility in resetting/deleting histograms
  - Will be moved to the WP for the next year
- Review support for writing same histogram/profile in a file several times (object versions) – DONE
- Organisation of third-party code (HDF5, expat, zlib) in externals/g4tools - DONE