Updates to Analysis Module

I. Hrivnacova
IJCLab Orsay (CNRS/IN2P3)

28th Geant4 Collaboration Meeting, Sapporo,
27 September 2023

Outline

- Updates in g4tools
  - No updates related to analysis category
- Updates in analysis
  - Developments in 11.1
  - Developments for 11.2
- 2023 Work plan items

Developments in 11.1

Design Changes

- Common implementation of **histogram handling** for all histogram/profiles type
  - 15 classes were reduced to 3
- 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` were implemented both in G4GenericAnalysisManager and the specific managers
  - This guaranties the identical behavior with the generic analysis manager

Object Cycles

- Added support for **writing the same histogram, profile and ntuple on file several times**
  - Requested at the G4 User forum
- 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

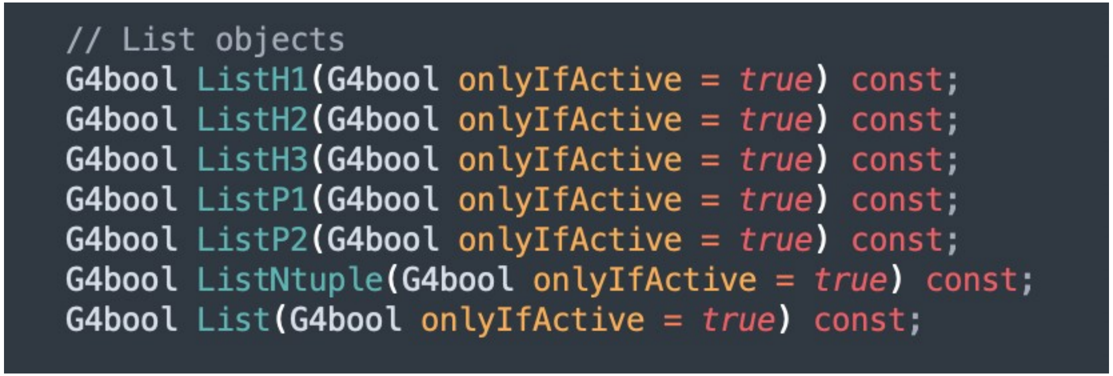
- The function `Write` can be now called more than once before `CloseFile`
- This leads to the need to provide UI commands for ”`open/close`” file and ”`reset, clear`” functions
- New example **analysis/AnaEx03**

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



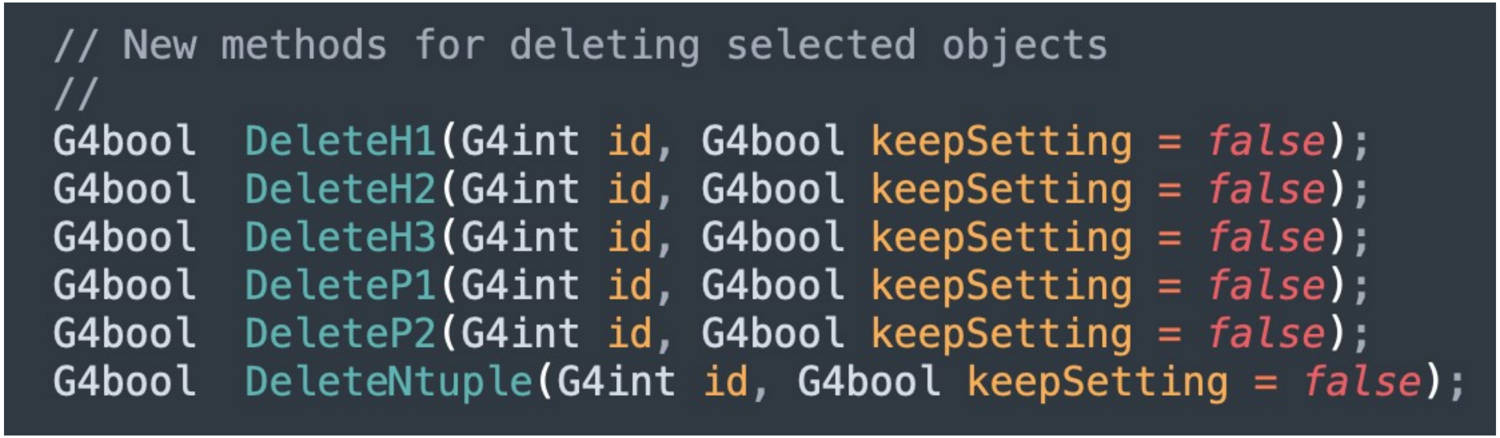
```
// List objects
G4bool ListH1(G4bool onlyIfActive = true) const;
G4bool ListH2(G4bool onlyIfActive = true) const;
G4bool ListH3(G4bool onlyIfActive = true) const;
G4bool ListP1(G4bool onlyIfActive = true) const;
G4bool ListP2(G4bool onlyIfActive = true) const;
G4bool ListNtuple(G4bool onlyIfActive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

- and corresponding UI commands:
  - `/analysis/xy/list [onlyIfActive],    xy = h1, h2, h3, p1, p2, ntuple`

New Developments for 11.2

Deleting Selected Objects

- Added G4AnalysisManger "`Delete`" functions
  - Requested several times at the G4 User forum



```
// New methods for deleting selected objects
//
G4bool  DeleteH1(G4int id, G4bool keepSetting = false);
G4bool  DeleteH2(G4int id, G4bool keepSetting = false);
G4bool  DeleteH3(G4int id, G4bool keepSetting = false);
G4bool  DeleteP1(G4int id, G4bool keepSetting = false);
G4bool  DeleteP2(G4int id, G4bool keepSetting = false);
G4bool  DeleteNtuple(G4int id, G4bool keepSetting = false);
```

- and corresponding UI commands:

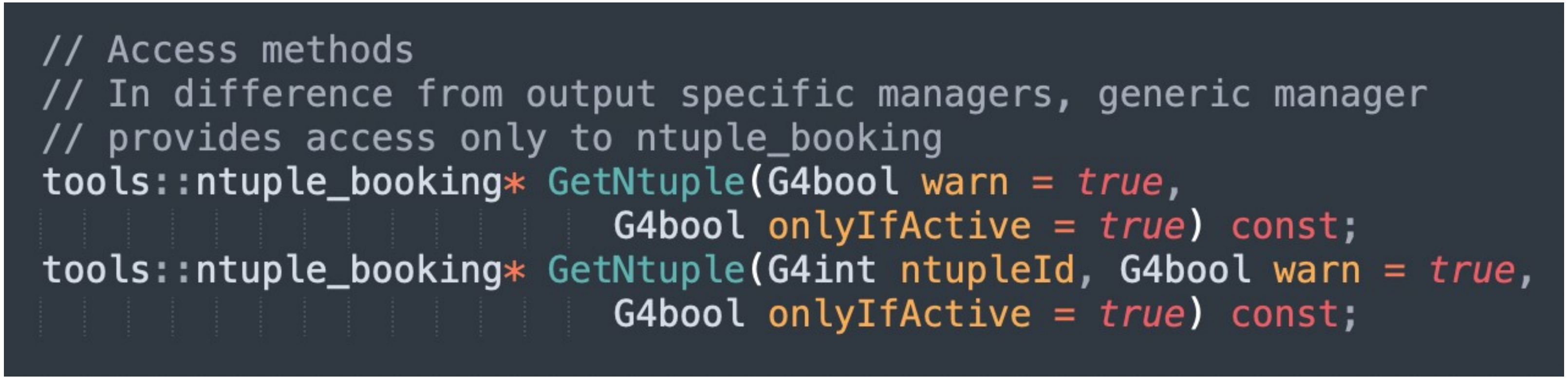
  -
    ```
    /analysis/xy/delete id [keepSetting],
                     xy = h1, h2, h3, p1, p2, ntuple
    ```

Deleting Selected Objects - 2

- The id’s of already created objects do not change with deleting selected objects
  - `GetHn|Pn|Ntuple(idx)` function will return `‘nullptr’` if the object at the `'idx'` position was deleted
    - It is under **user responsibility to test the returned value** before its use
- When an analysis manager function is called with `'idx'` of an object that was deleted, a warning is issued, but no segmentation violation error
- When a new object is created after some objects were deleted, it gets attributed **the first available Id**
  - Example:
    - Before Run 0: 5 h1 histograms are created
    - Before Run 1: the h1 histograms with `id = 2` and 4 are deleted
    - Before Run 2: two new histograms are created: they get `id = 2` and 4
  - If `'keepSetting'` was set to `'true'`, the additional information (object activation, object file name etc.) is kept after object deleting and it is automatically set to the new object created at this id

New “GetNtuple” Functions

- Added in G4GenericAnalysisManager:



```
// Access methods
// In difference from output specific managers, generic manager
// provides access only to ntuple_booking
tools::ntuple_booking* GetNtuple(G4bool warn = true,
                          G4bool onlyIfActive = true) const;
tools::ntuple_booking* GetNtuple(G4int ntupleId, G4bool warn = true,
                          G4bool onlyIfActive = true) const;
```

  - Currently `"GetNtuple"` is provided only in the output specific managers as the n-tuple type differs for each output format
  - After adding the possibility of deleting n-tuples, we need a function that allows to check the existence of an n-tuple
  - In difference from the output specific managers, the generic analysis manager returns `ntuple_booking`, that is common to all output types

New “ntuple” UI Commands

- Added UI commands for creating ntuple:
  - `/analysis/ntuple/create name title`
  - `/analysis/ntuple/create[I|F|D|S]Column name`
  - `/analysis/ntuple/finish`
- Issue #179 reported in GitLab
  - When using the default (`'Tasking'`) run manager in MT mode, the `/analysis/*/create` commands are executed twice
  - While with histograms this “just” creates a memory leak, **we get a break with Hdf5 when ntuples are added**, as HDF5 does not allow to create a group with the name that is already in use
    - The `test320` is using ‘MT’ run manager by default and the Hdf5 test is temorarily disabled in the tests with ‘TBB’ run manager (set via environment variable)

New example & test

- New extended example **analysis/AnaEx03**
  - Introduced already in 11.1 to demonstrate usage of analysis commands for file management, in particular writing histograms and ntuples in a file multiple times
  - In 11.2: added also histogram deleting & re-creating via UI commands
  - Ntuples will be added when the issue #179 is fixed
  - Only H1 histograms and n-tuples
- New test - **test320**
  - As test32, but with added deleting & re-creating of all objects types
  - More complete than AnaEx03 example:
    - Testing also `'keepSetting'` option and writing selected objects in another file(s) than the default one

Fixes

- Fixes in 11.1:
  - Fix in applying units in `"create"` and `"set"` UI commands.
  - Fixes in deleting files: do not delete (not empty) main n-tuple files with generic analysis manager, delete empty n-tuple csv and xml files created on the main thread. Fix in deleting files (root): do not delete the final (merged) file when only n-tuples are present.
  - Added a protection against `nullptr` in `G4VAnalysisManager::GetNofNtuples()`. Addressing problem report #2518.
  - Applied clang-tidy fixes.
- Fixes in 11.2:
  - Fixed creating histograms with user defined bins. Addressing bug report #2541.
  - Fixed implementation of set commands per dimension: `"hn|pn/setX|Y|Z"`. Addressing a problem report in Geant4 forum.
  - Fixed Coverity in `G4RootHnFileManager`

Plans

- 2023 Work plan item:
  - Investigate more flexibility in resetting/deleting analysis objects – (1)/(2) – DONE
- Beyond 2023
  - No new users requirements (except for a question about a possible support for more complex data structures like vector of vectors that we do not want to support)
  - Code maintenance