Analysis Tools in Geant4 10.2 and 10.3

Ivana Hřivnáčová

Institut de Physique Nucléaire (IPNO), Université Paris-Sud, CNRS-IN2P3, Orsay, France

E-mail: `ivana@ipno.in2p3.fr`

Guy Barrand

Laboratoire de l'Accélérateur Linéaire, Université Paris-Sud, CNRS-IN2P3, Orsay, France

E-mail: `barrand@lal.in2p3.fr`

Abstract. A new analysis category based on g4tools was added in Geant4 release 9.5 (2011 [1], [2], [3]) with the aim of providing users with a lightweight analysis tool available as part of the Geant4 installation without the need to link to an external analysis package. It has progressively be included in all Geant4 examples. Frequent questions in the Geant4 users forum show its increasing popularity in the Geant4 users community. In this presentation, we will give a brief overview of g4tools and the analysis category, then report on new developments since our CHEP 2013 contribution and on upcoming new features.

1. Geant4 Analysis

The classes in the analysis category provide a uniform, user-friendly interface to g4tools and hide the differences according to a selected output technology from the user. They take care of a higher-level management of the g4tools objects (files, histograms and ntuples), handle allocation and removal of the objects in memory and provide the access methods to them via indexes. They are fully integrated in the Geant4 framework: they follow Geant4 coding style and also implement the built-in Geant4 user interface commands that can be used by users to define or configure their analysis objects.

For a simplicity of use all user interface analysis functions are provided within a single class which is seen by the user as `G4AnalysisManager`. Internally, this type is defined via a `typedef` and it can point to one of four output type specific manager classes:

- `G4CsvAnalysisManager`
- `G4RootAnalysisManager`
- `G4XmlAnalysisManager`

All these manager classes implement the common interfaces, specific access functions (with a return type specific to the output format) and the singleton access method `Instance()`.

2. g4tools

The g4tools code is part of the inlib and exlib packages which had been originally developed to build highly portable visualisation and analysis applications running on standard laptop/desktop (Linux, MacOS, Windows) but also on smart devices (then iOS and Android).The code is mainly pure header and there is no config.h logic so that it is easy to install and embed. It is strongly OO. There is no implicit global management of instances and no writeable statics so that it is very modular and thread safe. See [3] and http://softinex.lal.in2p3.fr for more.

For Geant4, g4tools provides histograms, profiles and ntuples, and the IO code to write them in several supported file formats (ROOT, CSV, AIDA XML) and, when needed, also read them back from the files. We supported HBOOK for some time but this format had been deprecated since 10.3.

Since Geant4 10.2, we put in g4tools the graphical core of softinex, based on a scene graph logic, in order to permit batch plotting within Geant4. Only the postscript driver is included right now, then there is no GL-ES, X11, Windows, iOS, Android related code that comes with g4tools for the moment. For plotting a critical issue is text rendering. By default the g4tools plotting comes with the HPLOT fonts but much better rendering is possible by using the FreeType package that must be installed separately and is now known by the Geant4 cmake build system.

To help support multithreading and multiprocessing, code had been added to pass around histograms in MPI messages. Very recent developments include a way to parallelise the writing in one file of a ntuple at the ROOT format filled by parallel threads or processes, and this by arranging to pass basket/page of data in MPI messages.

3. New Features since Geant4 10.0

Geant4 analysis and g4tools presentations at the CHEP 2013 conference ([2], [3]) covered the functionalities available in Geant4 10.0. In this section we will present the major new developments since that time. A global view of the Geant4 analysis tools capabilities, including the new features presented in this section, is demonstrated in Fig. 1.

3.1. Histograms and Profiles

The set of supported histograms types was extended in Geant4 10.1 with three dimensional histograms and one and two dimensional profile histograms (profiles). Profile histograms (profiles) are used to display the mean value of Y and its error for each bin in X. The displayed error is by default the standard error on the mean (i.e. the standard deviation divided by the sqrt(n).)

Since this Geant4 version, histograms and profiles can be also written in the CSV file format, while in previous versions CSV could be used only with ntuples. Each histogram or profile is written in its own CSV output file, which name is generated automatically from the base file name.

3.2. Analysis Reader

The analysis reader classes were introduced in Geant4 10.1. They allow reading in g4tools objects from the files generated by the analysis manager(s) during processing Geant4 application.

The design of these classes follows the same pattern which is used for the manager classes: all user interface analysis reader functions are provided within a single class which is seen by the user as `G4AnalysisReader`. Internally, this type is defined via a `typedef` and it can point to one of four output type specific manager classes:

- `G4CsvAnalysisReader`
- `G4RootAnalysisReader`
- `G4XmlAnalysisReader`

While the histograms and profiles objects handled by the analysis reader are of the same type as those handled by the analysis manager, the ntuple type is different. The objects loaded with the analysis reader are handled independently from the objects created with analysis manager.

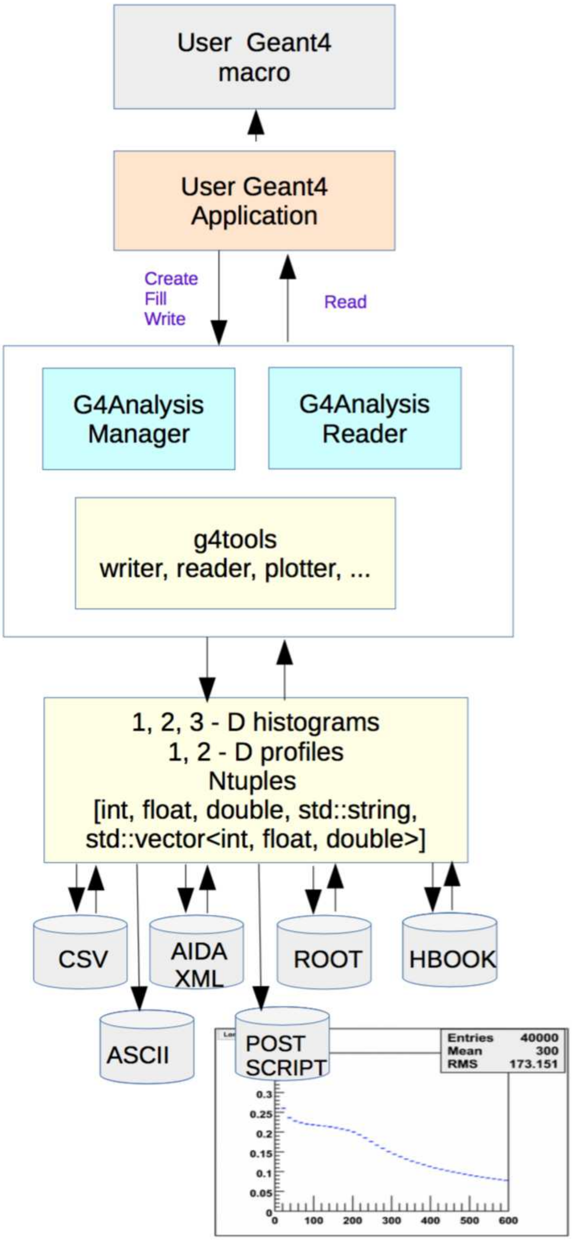

Figure 1. Overview of the Geant4 analysis tools capabilities.

3.3. "Batch plotting" facility

Since Geant4 10.2 version it is possible to produce a graphics output file in the Postscript format containing selected histograms and profiles. Users can activate plotting of selected objects using the `G4AnalysisManager` functions or the Geant4 user interface commands. The Postscript file is then created in addition to the standard output files.

User can choose from three plotting styles, two of which requires Geant4 libraries built with support for Freetype font rendering, and can also select the layout of the plots on the page, defined by the number columns and the number of rows in a page. The page size of the graphics output is fixed to A4 format. Finally, users can also customise the plot dimensions, which represent the plotter window size (width and height) in pixels. Opening more configuration parameters for users customisation can be considered in future according to the users feedback.

3.4. Accumulables

The classes for users "accumulables" management were added in 10.2 release for the purpose of simplification of users application code. The accumulables objects are named variables registered to the accumulable manager, which provides the access to them by name and performs their merging in multithreading mode. This classes set is independent from the analysis manager and reader classes.

To better reflect the meaning of these objects, the classes base name "Parameter" used in 10.2 was changed in "Accumulable" in 10.3. Further integration in the Geant4 framework is foreseen in the next Geant4 versions.

`G4Accumulable<T>` templated class can be used instead of built-in types in order to facilitate merging of the values accumulated on workers to the master thread. The `G4Accumulable<T>` object has, besides its value of the template type `T`, also a name, the initial value, which the value is set to in `Reset()` function and a merge mode, specifying the operation which is performed in `Merge()` function. The merge modes are defined in `G4MergeMode` class enumeration which supports addition, multiplication, maximum and minimum operation, the default operation is addition.

Users can also define their own accumulable class derived from `G4VAccumulable` abstract base class.

When running Geant4 application in multithreading mode, all accumulables registered to `G4AccumulableManager` accumulated on workers can be merged to the master thread by calling the `G4AccumulablesManager::Merge()` function. This step may be not necessary in future after a planned closer integration of the accumulable classes in the Geant4 kernel.

4. Multithreading and Multiprocessing

The analysis code was adapted for multithreading since the first Geant4 multithreading capable release 10.0. In multithreading mode, instances of the analysis manager are internally created on the master and worker threads and data accounting is processed in parallel on worker threads. The migration to multithreading did not require any changes in the user?s client analysis code.

Histograms produced on worker threads are automatically merged on the call to `Write()` and the result is written to a master file. Ntuples produced on worker threads are, by default, written to separate files, the names of which are generated automatically from a base file name, a thread identifier and eventually also an ntuple name.

Ntuples produced on thread workers are, by default, written on separate files, which names are generated automatically from a base file name, a thread identifier and eventually also an ntuple name. Since Geant4 version 10.3 it is possible to activate merging of ntuples with the ROOT output type. The ntuples produced on workers will be then progressively being merged to the main ntuples on the master. By default, the ntuples are written at the same file as the final histograms. Users can also select merging in a given number of files or customise the default basket size value. No merging of ntuples is provided with CSV and AIDA XML formats.

To simplify the scaling of a Geant4 application across nodes on a cluster Geant4 provides the support of MPI. In particular it is possible to run a hybrid MPI/MT application that uses MPI to scale across nodes and MT to scale across cores (see Fig. 2). This is demonstrated in the extended example parallel/MPI/exMPI03 which includes usage of Geant4 analysis.

The ability to send and receive histograms and profiles through MPI is available since Geant4 version 10.2 The `tools::histo::hmpi` interface to MPI, separated from the MPI-dependent implementation, `tools::mpi::hmpi`, allows to keep the analysis category in Geant4 independent from MPI libraries. The `G4MPIhistoMerger` class is provided in Geant4 extended/parallel/MPI example with no need to access the g4tools/analysis internals. Merging ntuples with ROOT output format over MPI workers is in the development plan for Geant4 10.4.

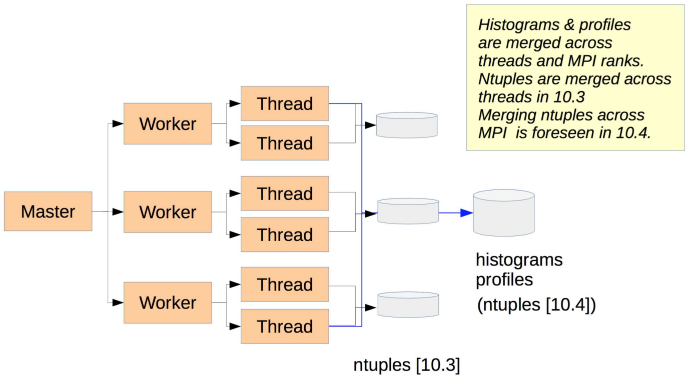
Histograms & profiles are merged across threads and MPI ranks.
Ntuples are merged across threads in 10.3
Merging ntuples across MPI is foreseen in 10.4.
Master
Worker
Worker
Worker
Thread
Thread
Thread
Thread
Thread
Thread
histograms
profiles
(ntuples [10.4])
ntuples [10.3]

Figure 2. Multithreading and multiprocessing capabilities in Geant4 10.3.

The g4analysis classes have proven to be thread-safe and no issues with concurrent processing were observed in Geant4 tests.

5. Conclusions

The analysis category has been included in Geant4 since five years. The migration of Geant4 examples to new analysis tools (replacing the usage of external tools based on AIDA (Abstract Interfaces for Data Analysis)) was accomplished in the 10.1 release and the activity in the analysis category of Geant4 user forum, shows a growing interest of both Geant4 and users. In this paper we gave an overview of its present status and reported on upcoming new features: new histogram object types, reading and batch plotting facilities, accumulables framework and new multithreading and multiprocessing capabilities.

References

[1] Agostinelli S et al 2003 *Nucl. Instrum, and Methods* **A506** 250-303
Allison J et al 2006 *IEEE Transactions on Nuclear Science* **53 No. 1** 270-278
Allison J et al 2016 *Nuclear Instruments and Methods in Physics Research* **A835** 186-225.
[2] Hrivnacova I 2014 *J. Phys.: Conf. Ser.* **513** 022014
[3] Barrand G 2014 *J. Phys.: Conf. Ser.* **513** 022002
[4] http://inexlib.lal.in2p3.fr
[5] http://root.cern.ch