

PAPER • OPEN ACCESS

Analysis Tools in Geant4 10.2 and 10.3

To cite this article: I Hivnáová and G Barrand 2017 *J. Phys.: Conf. Ser.* **898** 042018

Related content

- [Integration of g4tools in Geant4](#)
Ivana Hivnáová
- [A Virtual Geant4 Environment](#)
Go Iwai
- [The Geant4 physics validation repository](#)
H Wenzel, J Yarba and A Dotti

View the [article online](#) for updates and enhancements.

Analysis Tools in Geant4 10.2 and 10.3

I Hřivnáčová¹ and G Barrand²

¹ Institut de Physique Nucléaire, Université Paris-Sud, CNRS-IN2P3, Orsay, France

² Laboratoire de l'Accélérateur Linéaire, Université Paris-Sud, CNRS-IN2P3, Orsay, France

E-mail: ivana@ipno.in2p3.fr

Abstract. A new analysis category based on g4tools was added in Geant4 release 9.5 (2011). The aim was to provide 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 been 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. We report on new developments since our CHEP 2013 contribution as well as mention upcoming new features.

1. g4tools

The g4tools [1] code is part of the inlib and exlib [1] packages which were originally developed to build highly portable visualisation and analysis applications running on both laptop/desktop (Linux, MacOS, Windows) and mobile (then iOS and Android) operating systems. 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 object-oriented. There is no implicit global management of instances and no writable statics. Thus, it is very modular and thread safe (see [1] and [2] for more).

For Geant4 [3], g4tools provides histograms, profiles and ntuples, together with code to write them to several supported file formats (ROOT [4], CSV, AIDA XML) and, when needed, also read them back from the files. We supported HBOOK for some time but this format was dropped with 10.3.

Since Geant4 10.2, g4tools includes the graphical core of softinex [2], based on a scene graph logic, that permits plotting (histogram visualisation) within Geant4. Only a default postscript driver, which does not induce to attach to any external libraries, is included right now. PNG or JPEG files could be produced at the cost of a link toward externals, if there are user requests for these formats. There is also no GL-ES, X11, Windows, iOS or Android specific code included in g4tools at present to do this plotting on a screen. Nonetheless, code exists in softinex that would permit to plot for example in an OpenGL window created through the visualization system of Geant4. Text rendering, a key issue in plotting, uses the provided HPLOT fonts, but better rendering is possible using an optional interface to the FreeType [6] library which can be enabled when configuring and building Geant4.

To help support multithreading and multiprocessing, code has been added to pass around histograms in MPI messages. A recent development here uses the passing of baskets/pages of data to parallelise filling of an ntuple in a single ROOT format output file. The data are filled directly by parallel threads and via MPI messages by parallel processes.

2. Geant4 Analysis

The classes in the analysis category provide a uniform, user-friendly interface to g4tools and hide the differences from the user depending on which output technology was selected [5]. They take care of the 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's 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 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 three output type specific manager classes:

- **G4CsvAnalysisManager**
- **G4RootAnalysisManager**
- **G4XmlAnalysisManager**

Each class implements the common interfaces, specific access functions (with a return type specific to the output format) and the singleton access method **Instance()**.

3. New Features since Geant4 10.0

Geant4 analysis and g4tools presentations at the CHEP 2013 conference ([5], [1]) 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 shown in Fig. 1.

3.1. Histograms and Profiles

The set of supported histogram 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 $\text{sqrt}(n)$.)

Since this Geant4 version, histograms and profiles can also be written in the CSV file format, while in previous versions, CSV could only be used with ntuples. Each histogram or profile is written in its own CSV output file, the name of which 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 the run of a Geant4 application without the need to link to an external analysis package.

The design of these classes follows the same pattern 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 three 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.

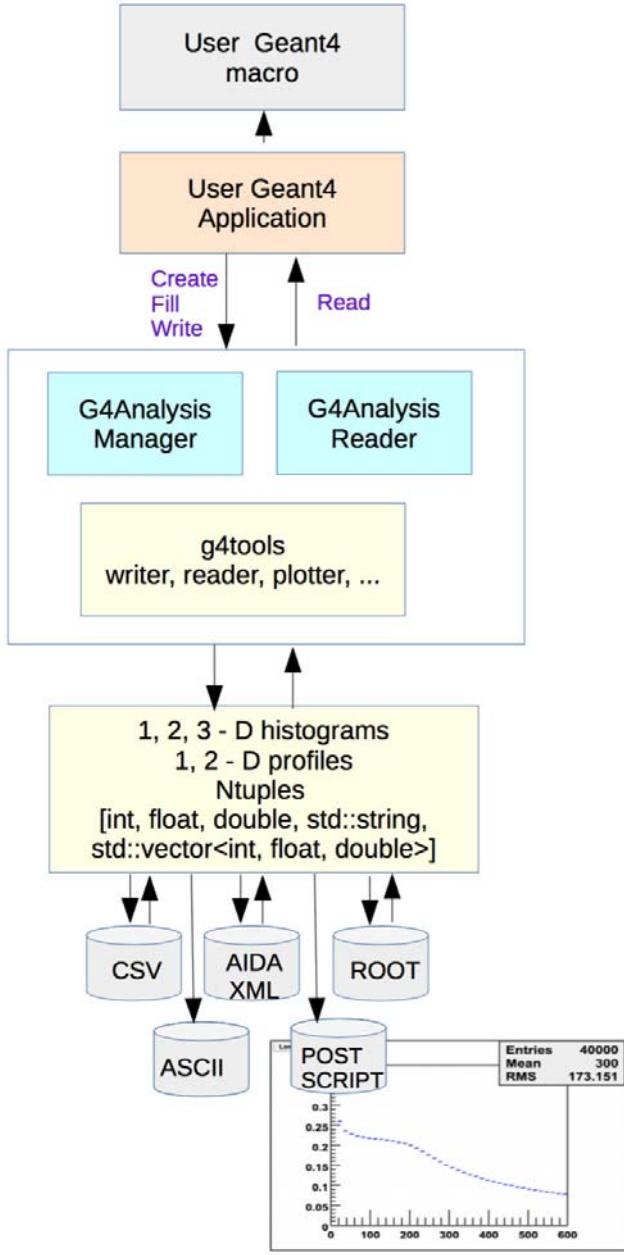


Figure 1. A global view of the Geant4 analysis tools capabilities.

3.3. Batch plotting facility

Since Geant4 10.2, 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. At present, only the Postscript format is available.

Users can choose from three plotting styles, two of which require the Geant4 library to be built with support for FreeType [6] font rendering. The user can also select the layout of the plots on the page, defined by the number of 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 customisation will be evaluated in the future according to the user's feedback.

3.4. Accumulables

Classes for managing "accumulables" were added in Geant4 10.2 to help simplify user's 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 set of classes is independent from the analysis manager and reader classes.

To better reflect the meaning of these objects, the word "Parameter" used in the class names in 10.2 was changed to "Accumulable" in 10.3. Further integration in the Geant4 framework is foreseen in future Geant4 releases.

`G4Accumulable<T>` template 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, an initial value and a merge mode. The initial value is set in the `Reset()` function. The merge mode specifies the operation performed in the `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 the `G4VAccumulable` abstract base class.

When running a multithreaded Geant4 application, all accumulables registered to `G4AccumulableManager` are accumulated on workers and can be merged to the master thread by calling the `G4AccumulablesManager::Merge()` function. This step may not be necessary in the 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 by the call to `Write()` and the result is written to a master file.

Ntuples produced on thread workers 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. With Geant4 version 10.3 it is possible to activate merging of ntuples with the ROOT output type. The ntuples produced on workers will then be progressively merged to the main ntuples on the master. By default, the ntuples are written to the same file as the final histograms. Users can also select merging into a given number of files or customise the default basket size value. Merging of ntuples with CSV and AIDA XML formats, not yet provided in 10.3, can be considered in the future versions according to the users feedback.

Geant4 provides the support of MPI to simplify the scaling of a Geant4 application across nodes on a cluster. 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 also exhibits the 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 keeping 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.

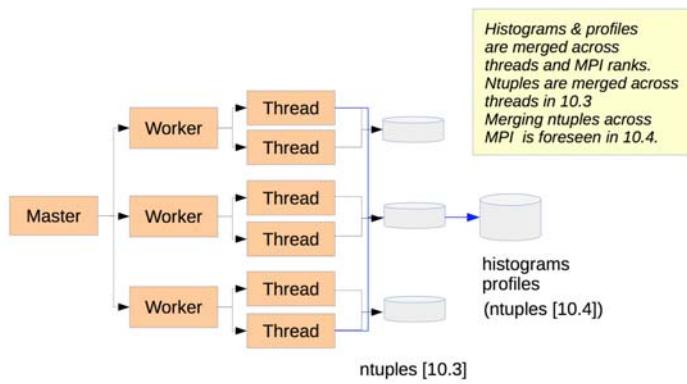


Figure 2. Multithreading and multiprocessing capabilities in Geant4 10.3.

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

5. Conclusion

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

References

- [1] Barrand G 2014 *J. Phys.: Conf. Ser.* **513** 022002
- [2] <http://softinex.lal.in2p3.fr>
- [3] 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.
- [4] <http://root.cern.ch>
- [5] Hrivnacova I 2014 *J. Phys.: Conf. Ser.* **513** 022014
- [6] <https://www.freetype.org>