

# Kernel I

Makoto Asai (Jefferson Lab)  
Geant4 Tutorial Course

# Contents



- General introduction and brief history
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



**GEANT4**  
A SIMULATION TOOLKIT



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



# Contents



- General introduction and brief history
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



**GEANT4**  
A SIMULATION TOOLKIT



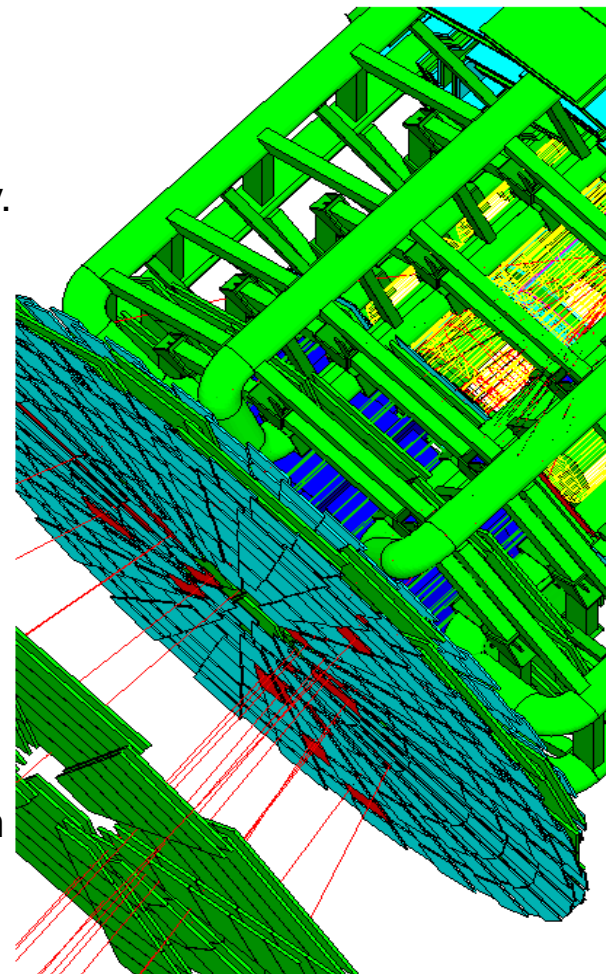
U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



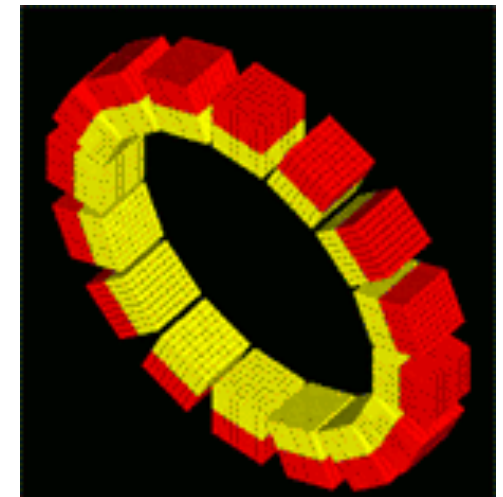
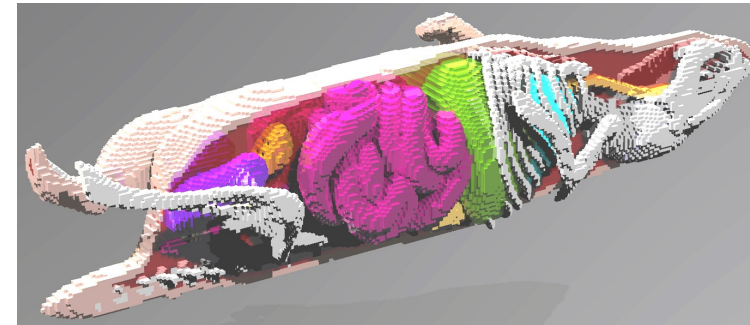
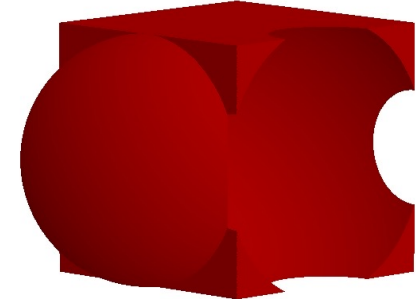
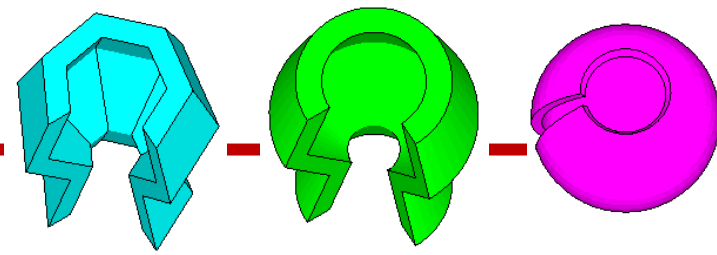
# Key Geant4 functionalities

- Geant4 is a general purpose Monte Carlo simulation tool for elementary particles passing through and interacting with matter. It finds quite a wide variety of user domains including high energy and nuclear physics, space engineering, medical applications, material science, radiation protection and security.
- Geant4 offers most, if not all, of the functionalities required for the simulation of elementary particle and nucleus passing through and interacting with matter.
  - Kernel
  - Geometry and navigation
  - Physics processes
  - Scoring
  - GUI and Visualization drivers
- Thanks to the polymorphism mechanism of C++, the users can easily plug-in their extensions without interfering with the other part of Geant4.
- Extensive user guide documents and examples are provided.



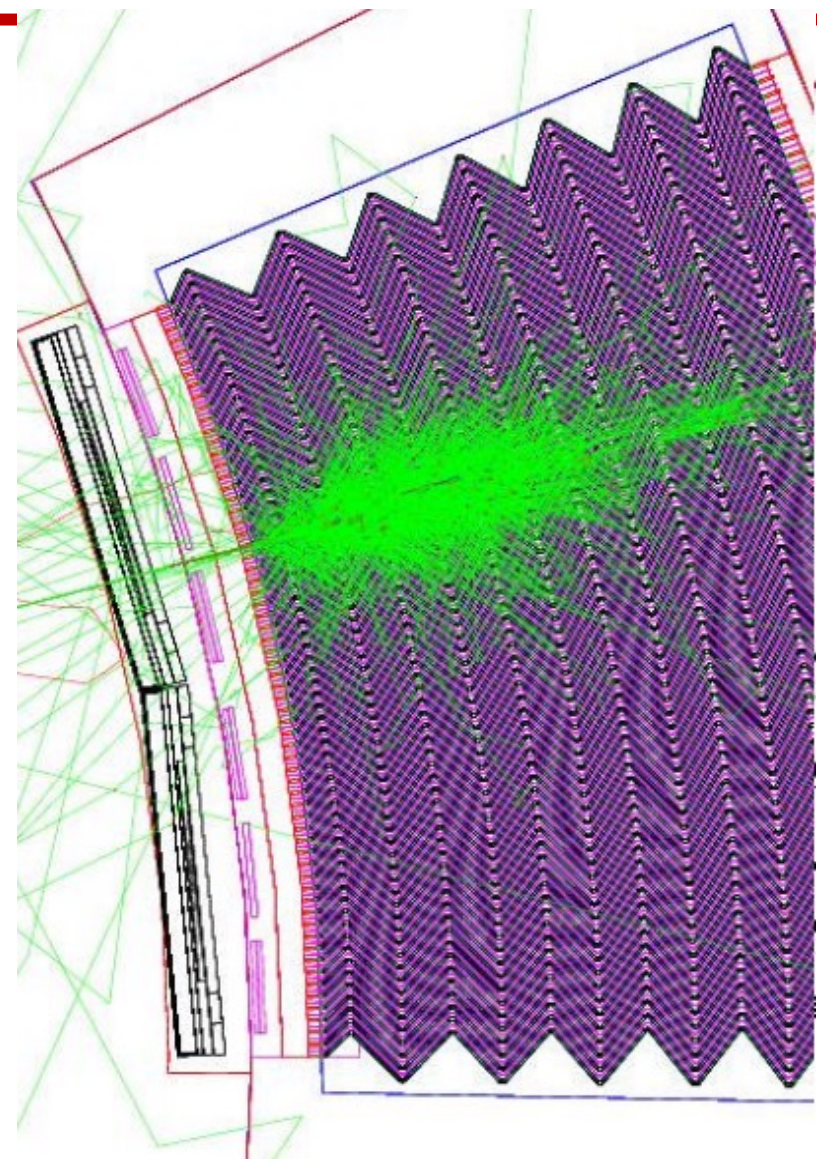
# Key geometry capabilities

- Richest collection of shapes
  - CSG (Constructed Solid Geometry), Boolean operation, Tessellated solid, etc.
  - The user can easily extend
- Describing a setup as hierarchy or ‘flat’ structure
  - Describing setups up to billions of volumes
  - Tools for creating & checking complex structures
  - Interface to CAD
- Navigating fast in complex geometry model
  - Automatic optimization
- Geometry models can be ‘dynamic’
  - Changing the setup at run-time, e.g. “moving objects”

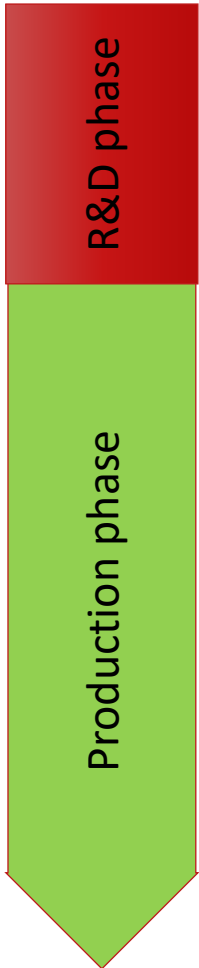


# Physics models in Geant4

- Geant4 offers
  - Electromagnetic processes
  - Hadronic and nuclear processes
  - Photon/lepton-hadron processes
  - Optical photon processes
  - Decay processes
  - Shower parameterization
  - Event biasing techniques
  - And you can plug-in more
- Geant4 provides sets of alternative physics models so that the user can freely choose appropriate models according to the type of his/her application.
  - For example, some models are more accurate than others at a sacrifice of speed.



# Geant4 History



- Early discussions even before having reliable C++ compilers.
- Dec '94 – R&D project start
- Apr '97 - First alpha release
- Jul '98 - First beta release
- Dec '98 - First Geant4 public release - version 1.0
- Several major architectural revisions
  - E.g. STL migration, “cuts per region”, parallel worlds, command-based scorer, **multithreading, task-based parallelization**
- Dec 8<sup>th</sup>, '23 – Geant4 version 11.2 release
  - Feb 16<sup>th</sup>, '24 - Geant4 11.2-patch01 release ← **Current version**
- We currently provide one public release every year.
  - Next version - Geant4 11.3 (planned on December 2024)

## Geant4

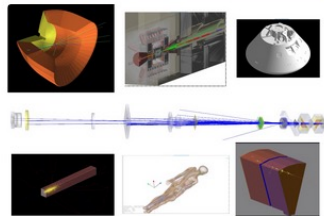
Toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science.

[Getting started](#)

### Get started

Everything you need to get started with Geant4.

[I'm ready to start!](#)



### About us

What is Geant4, where it's used, details on [Collaboration](#).

[Learn More](#)

### Download

Geant4 source code and installers are available for download, with source code under an [open source license](#).

Latest: [11.2.1](#)



### Collaboration

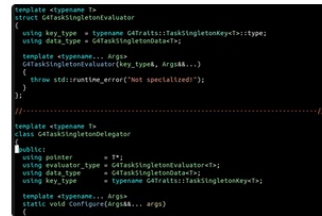
[Geant4 team and documents](#)

[Learn More](#)

### Docs

Documentation for Geant4, along with tutorials and guides, are available online.

[Read documentation](#)



### Contribute

How external users can contribute to Geant4.

[Learn More](#)

### News

[» More](#)

16 Feb 2024

[Release 11.2.1](#)

08 Dec 2023

[Release 11.2](#)

10 Nov 2023

[Release 11.1.3](#)

30 Jun 2023

[Release 11.2.beta](#)

19 Jun 2023

[Release 11.1.2](#)

### Events

[» More](#)

3/25/2024 - 3/29/2024 [Geant4 tutorial at JLAB](#), Jefferson Lab, Newport News (Virginia, USA)

3/25/2024 - 3/26/2024 [Geant4-DNA International tutorial](#), Osaka University, Osaka (Japan)

3/27/2024 - 3/29/2024 [The 5th Geant4 International User Conference at the Physics - Medicine - Biology frontier](#), Osaka University Nakanoshima Center, Osaka (Japan)



# Geant4 – a simulation toolkit



ScienceDirect

[https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)



Access through your institution

Purchase PDF

Article preview

Abstract

Introduction

Section snippets

References (125)

Cited by (23229)



Nuclear Instruments and Methods in Physics  
Research Section A: Accelerators, Spectrometers,  
Detectors and Associated Equipment

Volume 506, Issue 3, 1 July 2003, Pages 250-303



## GEANT4—a simulation toolkit

S. Agostinelli<sup>ae</sup>, J. Allison<sup>as</sup>, K. Amako<sup>e</sup>, J. Apostolakis<sup>a</sup>, H. Araujo<sup>aj</sup>, P. Arce<sup>lm xa</sup>,  
M. Asai<sup>g ai</sup>, D. Axen<sup>l t</sup>, S. Banerjee<sup>bi l</sup>, G. Barrand<sup>an</sup>, F. Behner<sup>l</sup>, L. Bellagamba<sup>c</sup>, J. Boudreau<sup>bd</sup>,  
L. Broglio<sup>af</sup>, A. Brunengo<sup>c</sup>, H. Burkhardt<sup>a</sup>, S. Chauvie<sup>bj bl</sup>, J. Chuma<sup>h</sup>, R. Chytracek<sup>a</sup>,  
G. Cooperman<sup>az</sup>...D. Zschesche<sup>af</sup>

Show more

+ Add to Mendeley Share Cite

[https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)

Get rights and content

### Abstract

GEANT4 is a toolkit for simulating the passage of particles through matter. It includes a complete range of functionality including tracking, geometry, physics models and hits. The physics processes offered cover a comprehensive range, including electromagnetic, hadronic and optical processes, a large set of long-lived particles, materials and elements, over a wide energy range starting, in some cases, from 250 eV and extending in others to the TeV energy range. It has been designed and constructed to expose the physics models utilised, to handle complex geometries, and to enable its easy adaptation for optimal use in different sets of applications. The toolkit is the result of a worldwide collaboration of physicists and software engineers. It has been created exploiting software engineering and object-oriented technology and implemented in the C++ programming language. It has been used in applications in particle physics, nuclear physics, accelerator design, space engineering and medical physics.

### Introduction

Modern particle and nuclear physics experiments pose enormous challenges in the creation of complex yet robust software frameworks and applications. Of particular

Recommended articles

Fully Geant4 compatible package for the simulation of Dark Matter in fixed target...

Computer Physics Communications, Volume 269, 2021...  
M. Bondi, ..., D. Shchukin

Application of Geant4 Monte Carlo simulation in dose calculations for small...

Medical Dosimetry, Volume 43, Issue 3, 2018, pp. 214-2...  
Somayeh Gholami, ..., Ali S. Meigooni

COLLIER: A fortran-based complex one-loop library in extended regularizations

Computer Physics Communications, Volume 212, 2017,...  
Ansgar Denner, ..., Lars Hofer

Show 3 more articles

Article Metrics

Citations

Citation Indexes:	18601
Patent Family Citations:	2
Policy Citations:	5

Captures

Readers:	2617
----------	------

Mentions

Blog Mentions:	4
News Mentions:	4
References:	7

Social Media

Shares, Likes & Comments: 10

Cited by (23234)

# Geant4 developments and applications

## Geant4 developments and applications

Publisher: IEEE

Cite This

PDF

J. Allison ; K. Amako ; J. Apostolakis ; H. Araujo ; P. Arce Dubois ; M. Asai ; G. Barrand ; R. Capra ; S. Chauvie ; R. Chytracsek ; G.A.P. Cirrone ; ... All Authors

4660

Cites in  
Papers

10

Cites in  
Patents

16404

Full  
Text Views



4660  
Cites in  
Papers

### Abstract:

Geant4 is a software toolkit for the simulation of the passage of particles through matter. It is used by a large number of experiments and projects in a variety of application domains, including high energy physics, astrophysics and space science, medical physics and radiation protection. Its functionality and modeling capabilities continue to be extended, while its performance is enhanced. An overview of recent developments in diverse areas of the toolkit is presented. These include performance optimization for complex setups; improvements for the propagation in fields; new options for event biasing; and additions and improvements in geometry, physics processes and interactive capabilities

Published in: IEEE Transactions on Nuclear Science ( Volume: 53 , Issue: 1, February 2006)

Page(s): 270 - 278

DOI: 10.1109/TNS.2006.869826

Date of Publication: February 2006

Publisher: IEEE

### ISSN Information:

III. Improvements in Detector Modeling

IV. Physics Extensions and Validation

V. Enhancement of Geant4 Interactive Capabilities

Show Full Outline ▾

### More Like This

Object oriented modelling of a recuperator process in stone wool production

Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces  
Published: 2010

The protection system for the superconducting elements of the Large Hadron Collider at CERN

Proceedings of the 1999 Particle Accelerator Conference (Cat. No.99CH36366)  
Published: 1999

Show More

# Recent developments in Geant4



<https://doi.org/10.1016/j.nima.2016.06.125>

View PDF Download full issue

## Outline

Abstract

Keywords

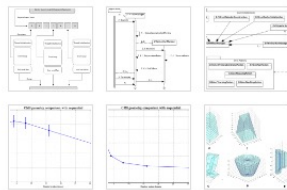
1. The evolution of GEANT4
2. Multithreading
3. Kernel functionalities
4. Recent developments in physics modeling
- 4.4. Results
5. Toolkit extensions
6. Validation
7. Outlook for the next decade

Acknowledgments

References

Show full outline

Cited by (3202)



Show 17 more figures

Tables (14)



Nuclear Instruments and Methods in Physics  
Research Section A: Accelerators, Spectrometers,  
Detectors and Associated Equipment

Volume 835, 1 November 2016, Pages 186-225

## Recent developments in GEANT4

J. Allison<sup>a,b</sup>, K. Amako<sup>c,a</sup>, J. Apostolakis<sup>d</sup>, P. Arce<sup>e</sup>, M. Asai<sup>f</sup>, T. Aso<sup>g</sup>, E. Bagli<sup>h</sup>, A. Bagulya<sup>i</sup>,  
S. Banerjee<sup>j</sup>, G. Barrand<sup>k</sup>, B.R. Beck<sup>l</sup>, A.G. Bogdanov<sup>m</sup>, D. Brandt<sup>n</sup>, J.M.C. Brown<sup>o</sup>, H. Burkhardt<sup>d</sup>,  
Ph. Canal<sup>j</sup>, D. Cano-Ott<sup>p</sup>, S. Chauvie<sup>q</sup>, K. Cho<sup>r</sup>, G.A.P. Cirrone<sup>s</sup>...H. Yoshida<sup>bs,a</sup>

Show more

+ Add to Mendeley Share Cite

<https://doi.org/10.1016/j.nima.2016.06.125>

Get rights and content

Under a Creative Commons license

open access

### Highlights

- Multithreading resulted in a smaller memory footprint and nearly linear speed-up.
- Scoring options, faster geometry primitives, more versatile visualization were added.
- Improved electromagnetic and hadronic models and cross sections were developed.
- Reverse Monte Carlo and general biasing methods were added.
- Physics validation efforts were expanded and new validation tools were added.

## Recommended articles

[Integral cross-sections of light nuclei in the Glauber-Gribov representation](#)

Nuclear Instruments and Methods in Physics Research Section A, Volume 835, Issue 1, November 2016, pp. 186-225  
V.M. Grichine

[Application of Geant4 Monte Carlo simulation in dose calculations for small...](#)

Medical Dosimetry, Volume 43, Issue 3, 2018, pp. 214-220  
Somayeh Gholami, ..., Ali S. Meigooni

[Latest Geant4 developments for PIXE applications](#)

Nuclear Instruments and Methods in Physics Research Section A, Volume 835, Issue 1, November 2016, pp. 186-225  
S. Bakr, ..., S. Guatelli

Show 3 more articles

## Article Metrics

### Citations

Citation Indexes: 2367  
Policy Citations: 1

### Captures

Readers: 789

### Mentions

References: 4

### Social Media

Shares, Likes & Comments: 25

View details

FEEDBACK



Kernel I - M. Asai (JLab)



# Contents



- General introduction and brief history
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



**GEANT4**  
A SIMULATION TOOLKIT



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



# Geant4 license

In response to user requests for clarification of Geant4's distribution policy, the collaboration announced a license.

It makes clear the user's wide-ranging freedom to use, extend or redistribute Geant4, even as part of some for-profit venture.

<https://geant4.web.cern.ch/download/license>

[About](#)[Download](#)[Documentation](#)[User Forum](#)[Bug Report](#)

[Home](#) > [Download](#) > [The Geant4 Software License](#)

## The Geant4 Software License

Established on 28 June 2006 for Geant4 release 8.1, subsequent patches and releases.

Previous releases are covered by the disclaimer included in the release. Copyright Holders of the Geant4 Collaboration.

Last revision: 28 June 2006

The collaboration has established the following list of institutions and individuals who hold copyright of parts of the Geant4 toolkit. In case there are attributions missing, the collaboration invites potential contributors to inform us of their contribution and who holds copyright over it, by writing to [license@geant4.org](mailto:license@geant4.org)

### Institutions

- Bath University, Bath, UK
- Budker Institute Nuclear Physics, Novosibirsk, Russia
- Budapest Technical University, Budapest, Hungary
- California Institute of Technology, Pasadena, USA
- CERN, European Organization for Nuclear Research, Geneva, Switzerland
- CIEMAT, Madrid, Spain
- CNRS-IN2P3, Institut National de Physique Nucleaire, France
- ESA, European Space Agency
- ETH, Zurich, Switzerland
- Fermi National Accelerator Laboratory, Batavia, USA
- Helsinki Institute of Physics, Helsinki, Finland
- IHEP, Protvino, Russia
- Imperial College, London, UK
- Institut fur experimentelle Kernphysic, Karlsruhe University, Karlsruhe, Germany
- Instituto de Fisica de Cantabria, Santander, Spain
- INFN, Istituto Nazionale di Fisica Nucleare, Italy
- IST National Institute for Cancer Research, Italy
- Jefferson Laboratory, USA

# Geant4 license

---

License has 8 points. The points are written clearly and simply.

1,2 and 3) tell the world who the software came from, and don't claim you are us.

Installation, use, reproduction, display, modification and redistribution of this software, with or without modification, in source and binary forms, are permitted on a non- exclusive basis. Any exercise of rights by you under this license is subject to the following conditions:

1. Redistributions of this software, in whole or in part, with or without modification, must reproduce the above copyright notice and these license conditions in this software, the user documentation and any other materials provided with the redistributed software.
2. The user documentation, if any, included with a redistribution, must include the following notice:  
"This product includes software developed by Members of the Geant4 Collaboration (<http://geant4.org>)."

If that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the modified version of this software itself.

3. The names "Geant4" and "The Geant4 toolkit" may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by [license@geant4.org](mailto:license@geant4.org). If this software is redistributed in modified form, the name and reference of the modified version must be clearly distinguishable from that of this software.

# Geant4 license

---

4) If you choose to give it away free to everyone, we can have it for free too.

5) You can't patent the parts we did.

4. You are under no obligation to provide anyone with any modifications of this software that you may develop, including but not limited to bug fixes, patches, upgrades or other enhancements or derivatives of the features, functionality or performance of this software. However, if you publish or distribute your modifications without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted all Members and all Copyright Holders of the Geant4 Collaboration a license to your modifications, including modifications protected by any patent owned by you, under the conditions of this license.

5. You may not include this software in whole or in part in any patent or patent application in respect of any modification of this software developed by you.

# Geant4 license

---

We don't claim that it works, and we're not responsible if it doesn't.

## 6. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE MEMBERS AND COPYRIGHT HOLDERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE MEMBERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE AND MODIFICATIONS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

## 7. LIMITATION OF LIABILITY

THE MEMBERS AND COPYRIGHT HOLDERS OF THE GEANT4 COLLABORATION AND CONTRIBUTORS SHALL HAVE NO LIABILITY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

8. This license shall terminate with immediate effect and without notice if you fail to comply with any of the terms of this license, or if you institute litigation against any Member or Copyright Holder of the Geant4 Collaboration with regard to this software.





# Contents



- General introduction and brief history
- Geant4 license
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



**GEANT4**  
A SIMULATION TOOLKIT



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



# Terminology (jargons)

---

- Run, event, track, step, step point
- Track  $\leftrightarrow$  trajectory, step  $\leftrightarrow$  trajectory point
- Process
  - At rest, along step, post step
- Cut = production threshold
- Sensitive detector, score, hit, hits collection,

# Run in Geant4

- As an analogy of the real experiment, a run of Geant4 starts with “**Beam On**”.
- Within a run, the user cannot change
  - detector setup
  - settings of physics processes
- Conceptually, a run is a collection of events which share the same detector and physics conditions.
  - **A run consists of one event loop.**
- At the beginning of a run, geometry is optimized for navigation and cross-section tables are calculated according to materials appear in the geometry and the cut-off values defined.
- **G4RunManager** class manages processing a run, a run is represented by **G4Run** class or a user-defined class derived from G4Run.
  - A run class may have a summary results of the run.
- **G4UserRunAction** is the optional user hook.

# Event in Geant4

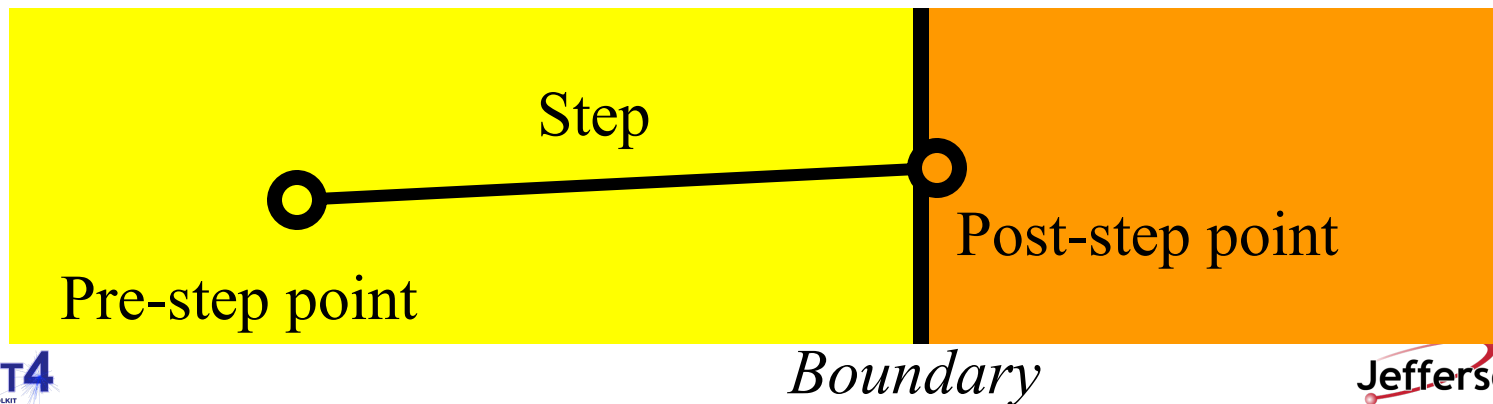
- An event is the basic unit of simulation in Geant4.
- At beginning of processing, primary tracks are generated. These primary tracks are pushed into a stack.
- A track is popped up from the stack one by one and “tracked”. Resulting secondary tracks are pushed into the stack.
  - This “tracking” lasts as long as the stack has a track.
- When the stack becomes empty, processing of one event is over.
- **G4Event** class represents an event. It has following objects at the end of its (successful) processing.
  - List of primary vertices and particles (as input)
  - Hits and Trajectory collections (as output)
- **G4EventManager** class manages processing an event. **G4UserEventAction** is the optional user hook.

# Track in Geant4

- Track is a **snapshot** of a particle.
  - It has physical quantities of **current instance** only. It does not record previous quantities.
  - **Step is a “delta” information to a track. Track is not a collection of steps. Instead, a track is being updated by steps.**
- Track object is deleted when
  - it goes out of the world volume,
  - it disappears (by e.g. decay, inelastic scattering),
  - it goes down to zero kinetic energy and no “AtRest” additional process is required, or
  - the user decides to kill it artificially.
- **No track object persists at the end of event.**
  - For the record of tracks, use trajectory class objects.
- **G4TrackingManager** manages processing a track, a track is represented by **G4Track** class.
- **G4UserTrackingAction** is the optional user hook.

# Step in Geant4

- Step has two points and also “delta” information of a particle (energy loss on the step, time-of-flight spent by the step, etc.).
- Each point knows the volume (and material). In case a step is limited by a volume boundary, the end point physically stands on the boundary, and it **logically belongs to the next volume**.
  - Because one step knows materials of two volumes, boundary processes such as transition radiation or refraction could be simulated.
- **G4SteppingManager** class manages processing a step, a step is represented by **G4Step** class.
- **G4UserSteppingAction** is the optional user hook.



# Trajectory and trajectory point

- Track does not keep its trace. No track object persists at the end of event.
- **G4Trajectory** is the class which copies some of G4Track information.  
**G4TrajectoryPoint** is the class which copies some of G4Step information.
  - G4Trajectory has a vector of G4TrajectoryPoint.
  - At the end of event processing, G4Event has a collection of G4Trajectory objects.
    - /tracking/storeTrajectory must be set to 1.
- Keep in mind the distinction.
  - G4Track  $\leftrightarrow$  G4Trajectory, G4Step  $\leftrightarrow$  G4TrajectoryPoint
- Given G4Trajectory and G4TrajectoryPoint objects persist till the end of an event, you should be careful not to store too many trajectories.
  - E.g. avoid for high energy EM shower tracks.
- G4Trajectory and G4TrajectoryPoint store only the minimum information.
  - You can create your own trajectory / trajectory point classes to store information you need. G4VTrajectory and G4VTrajectoryPoint are base classes.

# Particle in Geant4

- A particle in Geant4 is represented by three layers of classes.
- **G4Track**
  - Position, geometrical information, etc.
  - This is a class representing a particle to be tracked.
- **G4DynamicParticle**
  - "Dynamic" physical properties of a particle, such as momentum, energy, spin, etc.
  - Each G4Track object has its own and unique G4DynamicParticle object.
  - This is a class representing an individual particle.
- **G4ParticleDefinition**
  - "Static" properties of a particle, such as charge, mass, life time, decay channels, etc.
  - G4ProcessManager which describes processes involving to the particle
  - All G4DynamicParticle objects of same kind of particle share the same G4ParticleDefinition.



# Tracking and processes

---

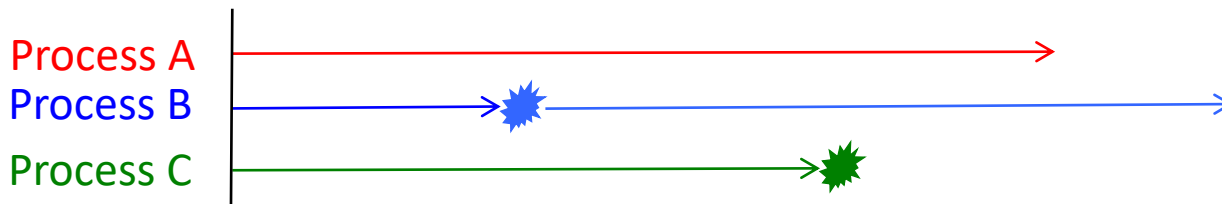
- Geant4 tracking is general.
  - It is independent to
    - the particle type
    - the physics processes involving to a particle
  - It gives the chance to all processes
    - To contribute to determining the step length
    - To contribute any possible changes in physical quantities of the track
    - To generate secondary particles
    - To suggest changes in the state of the track
      - e.g. to suspend, postpone or kill it.

# Processes in Geant4

- In Geant4, “process” is an abstract concept that may affect any of the track data.
  - Momentum, energy, position, secondary generation, fate of the track, etc.
- Particle transportation is a process as well, by which a particle interacts with geometrical volume boundaries and field of any kind.
  - Because of this, shower parameterization process can take over from the ordinary transportation without modifying the transportation process.
- Each particle has its own list of applicable processes. At each step, all processes listed are invoked to get proposed physical interaction lengths.
- The process which requires the shortest interaction length (in space-time) limits the step.
- Each process has one or combination of the following natures.
  - AtRest
    - e.g. muon decay at rest
  - AlongStep (a.k.a. continuous process)
    - e.g. Celenkov process
  - PostStep (a.k.a. discrete process)
    - e.g. decay on the fly

# Process competition

- “Ordinary” physics makes point-like interaction. Given many physics processes have chances to occur, one needs to make a fair competition among these eligible processes.
- Given PDF of each process, one can sample the path length **normalized by mean free path** (radiation length, hadronic interaction length, decay time, etc.) for each physics process.
- Compare the path lengths proposed by all physics processes. The process that proposes the shortest length occurs.
  - Given the length is normalized, competition should be made by the actual length ( normalized length x mean free path of the material ).
- Once the particle experiences an interaction by a physics process, the path length for that process is re-sampled, while proposed path lengths of other processes are reduced by the length traveled.
- Continuous processes (continuous energy loss, multiple scattering, Cherenkov radiation, etc.) are applied cumulatively.



# Cuts in Geant4

---

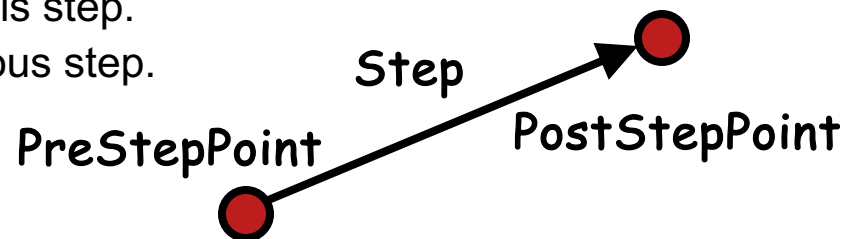
- A Cut in Geant4 is a **production threshold**.
  - Not tracking cut, which does not exist in Geant4 as default.
    - **All tracks are traced down to zero kinetic energy.**
  - It is applied **only** for physics processes that have infrared divergence
- Much detail will be discussed at later talks on physics.

# Track status

- At the end of each step, according to the processes involved, the state of a track may be changed.
  - The user can also change the status in **UserSteppingAction**.
  - Statuses shown in **green** are artificial, i.e. Geant4 kernel won't set them, but the user can set.
- fAlive
  - Continue the tracking.
- fStopButAlive
  - The track has come to zero kinetic energy, but still AtRest process to occur.
- fStopAndKill
  - The track has lost its identity because it has decayed, interacted or gone beyond the world boundary.
  - Secondaries will be pushed to the stack.
- **fKillTrackAndSecondaries**
  - Kill the current track and also associated secondaries.
- **fSuspend**
  - Suspend processing of the current track and push it and its secondaries to the stack.
- **fSuspendAndWait**
  - Suspend processing of the current track and force it into the waiting stack.
- **fPostponeToNextEvent**
  - Postpone processing of the current track to the next event.
  - Secondaries are still being processed within the current event.

# Step status

- Step status is attached to **G4StepPoint** to indicate why that particular step was determined.
  - Use “**PostStepPoint**” to get the status of this step.
  - “**PreStepPoint**” has the status of the previous step.
  - **fWorldBoundary**
    - Step reached the world boundary
  - **fGeomBoundary**
    - Step is limited by a volume boundary except the world
  - **fAtRestDoltProc**, **fAlongStepDoltProc**, **fPostStepDoltProc**
    - Step is limited by a **AtRest**, **AlongStep** or **PostStep** process
  - **fUserDefinedLimit**
    - Step is limited by the user Step limit
  - **fExclusivelyForcedProc**
    - Step is limited by an exclusively forced (e.g. shower parameterization) process
  - **fUndefined**
    - Step not defined yet
- If you want to identify **the first step in a volume**, pick **fGeomBoundary** status in **PreStepPoint**.
- If you want to identify **a step getting out of a volume**, pick **fGeomBoundary** status in **PostStepPoint**



# Extract useful information

- Given geometry, physics and primary track generation, Geant4 does proper physics simulation “silently”.
  - You have to do something to **extract information useful to you**.
- There are three ways:
  - Built-in scoring commands
    - Most commonly-used physics quantities are available.
  - Use scorers in the tracking volume
    - Create scores for each event
    - Create own Run class to accumulate scores
  - Assign **G4VSensitiveDetector** to a volume to generate “hit”.
    - Use user hooks (G4UserEventAction, G4UserRunAction) to get event / run summary
- You may also use user hooks (G4UserTrackingAction, G4UserSteppingAction, etc.)
  - You have full access to almost all information
  - Straight-forward in sequential mode, but do-it-yourself

# Unit system

- Internal unit system used in Geant4 is completely hidden not only from user's code but also from Geant4 source code implementation.

- Each hard-coded number must be multiplied by its proper unit.

```
radius = 10.0 * cm;
```

```
kineticE = 1.0 * GeV;
```

- To get a number, it must be divided by a proper unit.

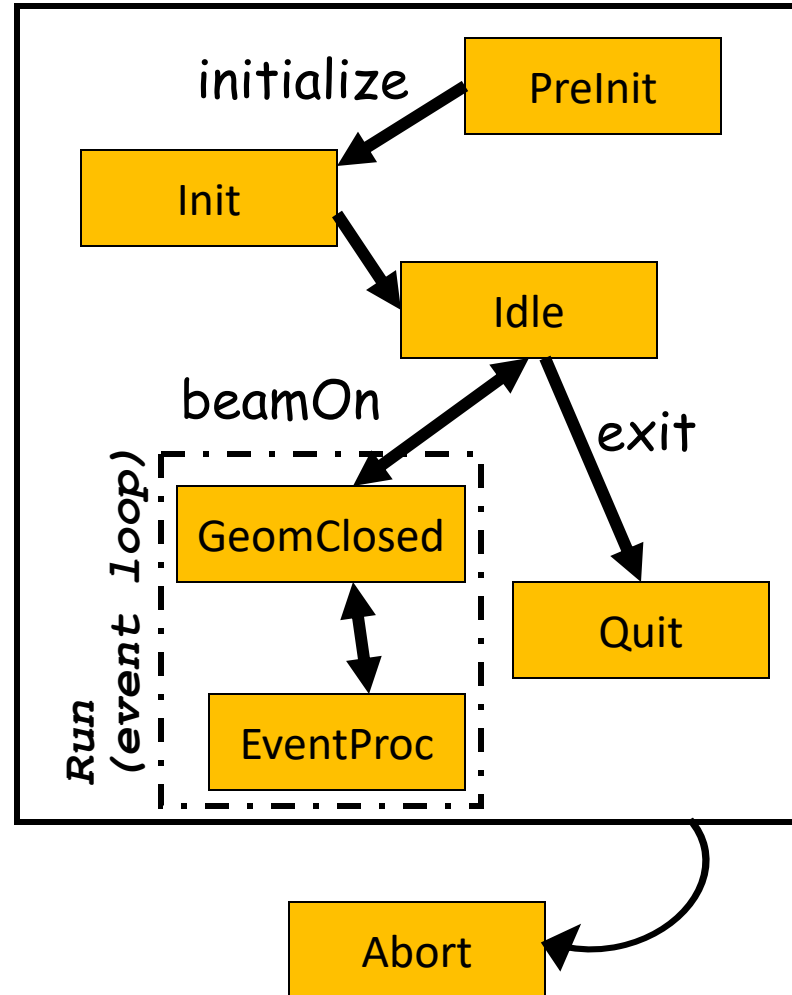
```
G4cout << eDep / MeV << " [MeV]" << G4endl;
```

- Most of commonly used units are provided and user can add his/her own units.
- By this unit system, source code becomes more readable and importing / exporting physical quantities becomes straightforward.
  - For particular application, user can change the internal unit to suitable alternative unit without affecting to the result.



# Geant4 as a state machine

- Geant4 has seven application states.
  - G4State\_PreInit
    - Initial condition
  - G4State\_Init
    - During initialization
  - G4State\_Idle
    - Ready to start a run
  - G4State\_GeomClosed
    - Geometry is optimized and ready to process an event
  - G4State\_EventProc
    - An event is processing
  - G4State\_Quit
    - (Normal) termination
  - G4State\_Abort
    - A fatal exception occurred and program is aborting



Note: Toggles between *GeomClosed* and *EventProc* occur for each thread asynchronously in multithreaded mode.

# G4cout, G4cerr

- **G4cout** and **G4cerr** are *ostream* objects defined by Geant4.
  - **G4endl** is also provided.

```
G4cout << "Hello Geant4!" << G4endl;
```

- Some GUIs are buffering output streams so that they display print-outs on another window or provide storing / editing functionality.
  - The user should not use `std::cout`, etc.
- The user should not use `std::cin` for input. Use user-defined commands provided by intercoms category in Geant4.
  - Ordinary file I/O is OK.

# Contents



- General introduction and brief history
- Geant4 kernel
  - Basic concepts and kernel structure
  - User classes



**GEANT4**  
A SIMULATION TOOLKIT



U.S. DEPARTMENT OF  
**ENERGY**

Office of  
Science



# To use Geant4, you have to...

- Geant4 is a toolkit. You have to build an application.
- To make an application, you have to
  - Define your geometrical setup
    - Material, volume
  - Define physics to get involved
    - Particles, physics processes/models
    - Production thresholds
  - Define how an event starts
    - Primary track generation
  - Extract information useful to you
- You may also want to
  - Visualize geometry, trajectories and physics output
  - Utilize (Graphical) User Interface
  - Define your own UI commands
  - etc.

# User classes

- **main()**
  - Geant4 does not provide *main()*.
- Initialization classes
  - Use `G4RunManager::SetUserInitialization()` to define.
  - Invoked at the initialization
    - **G4VUserDetectorConstruction**
    - **G4VUserPhysicsList**
    - **G4VUserActionInitialization**
- Action classes
  - Instantiate in your `G4VUserActionInitialization`.
  - Invoked during an event loop
    - **G4VUserPrimaryGeneratorAction**
    - `G4UserRunAction`
    - `G4UserEventAction`
    - `G4UserStackingAction`
    - `G4UserTrackingAction`
    - `G4UserSteppingAction`

Note : classes written in **red** are mandatory.

# The main program

- Geant4 does not provide a *main()*.
- In your *main()*, you have to
  - Construct G4RunManager
    - **G4RunManagerFactory::CreateRunManager()** instantiates a RunManager of your choice.
      - “Tasking” : G4TaskingRunManager (tasking mode – default)
      - “TBB” : G4TBBRunManager (tasking mode with TBB)
      - “MT” : G4MTRunManager (multithreaded mode based on p-thread)
      - “Serial” : G4Runmanager (sequential mode)
    - Environment variables may overwrite the choice.
      - G4RUN\_MANAGER\_TYPE can be set to override the "default"
        - If the requested type isn't available, then it will fall back to the system default
      - G4FORCE\_RUN\_MANAGER\_TYPE can be set to force a specific type
        - A G4Exception is raised if the requested type is not available
    - Set user mandatory initialization classes to RunManager
      - G4VUserDetectorConstruction
      - G4VUserPhysicsList
      - G4VUserActionInitialization
  - You can define VisManager, (G)UI session, optional user action classes, and/or your persistency manager in your *main()*.

# Describe your detector

- Derive your own concrete class from **G4VUserDetectorConstruction** abstract base class.
- In the virtual method *Construct()*, that is invoked in the master thread (and in sequential mode)
  - Instantiate all necessary materials
  - Instantiate volumes of your detector geometry
- In the virtual method *ConstructSDandField()*, that is invoked in each worker thread (and in sequential mode)
  - Instantiate your sensitive detector classes and field classes and set them to the corresponding logical volumes and field managers, respectively.

# Select physics processes

- Geant4 does not have any default particles or processes.
  - Even for the particle transportation, you have to define it explicitly.
- Derive your own concrete class from **G4VUserPhysicsList** abstract base class.
  - Define all necessary particles
  - Define all necessary processes and assign them to proper particles
  - Define cut-off ranges applied to the world (and each region)
- Primarily, the user's task is choosing a “pre-packaged” physics list, that combines physics processes and models that are relevant to a typical application use-cases.
  - If “pre-packaged” physics lists do not meet your needs, you may add or alternate some processes/models.
  - If you are brave enough, you may implement your own physics list.



# Generate primary event

- This is the only mandatory user action class.
- Derive your concrete class from **G4VUserPrimaryGeneratorAction** abstract base class.
- Pass a G4Event object to one or more primary generator concrete class objects which generate primary vertices and primary particles.
- Geant4 provides several generators in addition to the G4VPrimaryParticlegenerator base class.
  - G4ParticleGun
  - G4HEPEvtInterface, G4HepMCInterface
    - Interface to /hepevt/ common block or HepMC class
  - G4GeneralParticleSource
    - Define radioactivity

# Optional user action classes

- All user action classes, methods of which are invoked during “Beam On”, must be constructed in the user’s *main()* and must be set to the RunManager.
- **G4UserRunAction**
  - G4Run\* GenerateRun()
    - Instantiate user-customized run object
  - void BeginOfRunAction(const G4Run\*)
    - Define histograms
  - void EndOfRunAction(const G4Run\*)
    - Analyze the run
    - Store histograms
- **G4UserEventAction**
  - void BeginOfEventAction(const G4Event\*)
    - Event selection
  - void EndOfEventAction(const G4Event\*)
    - Output event information

# Optional user action classes

- **G4UserStackingAction**

- void PrepareNewEvent()

- Reset priority control

- G4ClassificationOfNewTrack ClassifyNewTrack(const G4Track\*)

- Invoked every time a new track is pushed

- Classify a new track -- priority control

- Urgent, Waiting, PostponeToNextEvent, Kill

- void NewStage()

- Invoked when the Urgent stack becomes empty

- Change the classification criteria

- Event filtering (Event abortion)

# Optional user action classes

- **G4UserTrackingAction**

- void PreUserTrackingAction(const G4Track\*)

- Decide trajectory should be stored or not
    - Create user-defined trajectory

- void PostUserTrackingAction(const G4Track\*)

- Delete unnecessary trajectory

- **G4UserSteppingAction**

- void UserSteppingAction(const G4Step\*)

- Kill / suspend / postpone the track

# Instantiate user action classes

- **G4VUserActionInitialization** has two virtual methods.
- *Build()*
  - Invoked at the beginning of each worker thread as well as in sequential mode
  - Use *SetUserAction()* method to register pointers of all user actions.
  - In multithreaded mode, all user action class objects instantiated in this method are thread-local.
    - User run action instantiated in this method is for thread-local run
- *BuildForMaster()*
  - Invoked only at the beginning of the master thread in multithreaded mode
  - Use *SetUserAction()* method to register pointer of user run action for the global run.

# Let me remind you...

- Instantiate RunManager and set initialization classes (and optional vis/GUI/etc.)
  - *main()*  
Hands-on
- Define material and geometry
  - G4VUserDetectorConstruction  
Material and Geometry lectures
- Select appropriate particles and processes and define production threshold(s)
  - G4VUserPhysicsList  
Physics lectures
- Instantiate user action classes
  - G4VUserActionInitialization  
Hands-on
- Define the way of primary particle generation
  - G4VUserPrimaryGeneratorAction  
Primary particle lecture
- Define the way to extract useful information from Geant4
  - G4VUserDetectorConstruction, G4UserEventAction, G4Run, G4UserRunAction
  - G4SensitiveDetector, G4VHit, G4VHitsCollection  
Scoring lectures