



Geant4 Develop, Build and Test Tools Update

Ben Morgan



WARWICK
THE UNIVERSITY OF WARWICK

GitLab/Hub Topics



GitLab Status

- Approaching 4000 MRs in total, ~450 since 11.1, plus those for patches to 11.1/0
- CERN GitLab updated to 16.0
- Some ongoing GitLab <=> Jenkins communication issues to resolve with CERN IT.
- Ongoing work to streamline infrastructure and provide Tech Note

Search GitLab

G4 geant4 > G4 geant4-dev > Merge requests

Open 6 Merged 427 Closed 25 All 458

Recent searches ▼ Milestone = "%Release 11.2" ×

OpNovice2-V11-01-00: add option to kill photon after boundary interaction
!3914 · created 17 hours ago by Daren Sawkey 🕒 Release 11.2

Draft: track-V11-01-01, event-V11-01-02 and tracking-V11-01-03: Introduce new status: fSuspend
!3866 · created 3 weeks ago by John Allison 🕒 Release 11.2
Waiting for approval

event-V11-01-02, run-V11-01-06, global-V11-01-16: Introducing G4SubEvent and related classes.
!3897 · created 1 week ago by Makoto Asai 🕒 Release 11.2
Staged

materials-V11-01-04: hadr-lepnuc-V11-01-00: emutils-V11-01-14: hadr-cross-V11-01-10: introduce
!3910 · created 2 days ago by Vladimir Ivantchenko 🕒 Release 11.2
Staged

radiobiology-V11-01-00 and ctests-V11-01-08: implementation of new extended example
!3800 · created 2 months ago by Pablo Cirrone 🕒 Release 11.2
Requiring fixes Waiting for rebase

Draft: Boris-SDC integration method
!3029 · created 11 months ago by Divyansh Tiwari 🕒 Release 11.2

GitLab TODO Items: *Help Required*

- **GitLab/CI maintenance, enhancement, documentation is low on FTE**
 - *It's not particularly glamorous, but critical to maintaining an efficient development cycle!*
- **Priority tasks:**
 - *Implement Nightly builds for patch branches to simplify/unify with “main”. Make patches branch in “always ready to validate” state, so patches can be more frequently/easily released*
 - *Only launch CI builds when tested code changes*
 - *Test install+use of install in Nightlies*
- **One longer term item is to review GitLab CI/Runners to replace Jenkins**
- **Also always in need of Shifters to monitor and process Merge Requests!**
- ***If you can contribute to long term support, or the above tasks, then please contact Software Management!***

G4 geant4-dev

- Project information
- Repository
- Files**
- Commits
- Branches
- Tags
- Contributors
- Graph
- Compare
- Locked Files
- Issues 36
- Merge requests 14
- CI/CD
- Security & Compliance
- Deployments
- « Collapse sidebar

CONTRIBUTING.rst 51.8 KB

</> Edit Web IDE Lock Replace Delete

Contributing to Geant4

The authoritative Geant4 Git repository is hosted on the [CERN GitLab](https://gitlab.cern.ch/geant4/geant4-dev) instance at <https://gitlab.cern.ch/geant4/geant4-dev>, accessible to members of the [Geant4 Collaboration](#). Contributions to Geant4 from outside the Collaboration are welcome, subject to successful testing and validation, via pull requests submitted through our GitHub mirror of published releases hosted at <https://github.com/Geant4/geant4>.

If you want to try out the Workflow documented below, the <https://gitlab.cern.ch/geant4/geant4-dev> repository provides a "sandbox" which can be used for experimentation.

If you have any issues with or questions on the workflow, please start an issue on the Issues Board at <https://gitlab.cern.ch/geant4/geant4-dev/issues>.

Basic Tools

Git

Git is the version control system used by Geant4. If it is not installed on your system already, it is usually available through package managers for your system (apt, yum, Homebrew, vcpkg) or comes bundled with IDEs such as XCode or Visual Studio. Failing this, the Git homepage provides a range of [binary bundles for Linux, MacOS, and Windows](#).

If you are new to git or need additional information, a wide range of documentation and tutorials are available, including:

- The [Atlassian Tutorials](#) are a good starting point, and also cover more advanced topics.
- The official [Pro Git book](#) offers greater depth and breadth. Chapters 2 and 3 covering the basics.

Merge Request Process Documentation

[A living document](#), so Issues/MRs to improve it are very welcome



Benjamin Morgan @bmorgan started a thread on an old version of the diff 1 week ago

Hide thread

Resolved 1 week ago by Makoto Asai

source/event/include/G4Event.hh

```

228 231 // Flag to keep the event until the end of run
229 232 G4bool keepTheEvent = false;
230 233 mutable G4int grips = 0;
234 +
235 + //===== for sub-event parallelism
236 +
237 + public:
238 + G4int SpawnSubEvent(G4SubEvent*);
239 + G4int TerminateSubEvent(G4SubEvent*);
240 + inline G4int GetNumberOfRemainingSubEvents() const
241 + { return fSubEvtVector.size(); }

```



Benjamin Morgan @bmorgan · 1 week ago

Owner



Suggested change

Apply suggestion

```

241 - { return fSubEvtVector.size(); }
241 + { return (G4int)fSubEvtVector.size(); }

```

Needed to avoid compiler warnings on some platforms.

- Makoto Asai changed this line in [version 2 of the diff](#) 1 week ago · [Compare changes](#)

Assignee

Edit



Gabriele Cosmo

Reviewer

Edit



Benjamin Morgan



Labels

Edit

Staged

Milestone

Edit

Release 11.2

Time tracking

+

No estimate or time spent

Lock merge request

Edit

Unlocked

Notifications



5 Participants

Reminder on MR Reviews

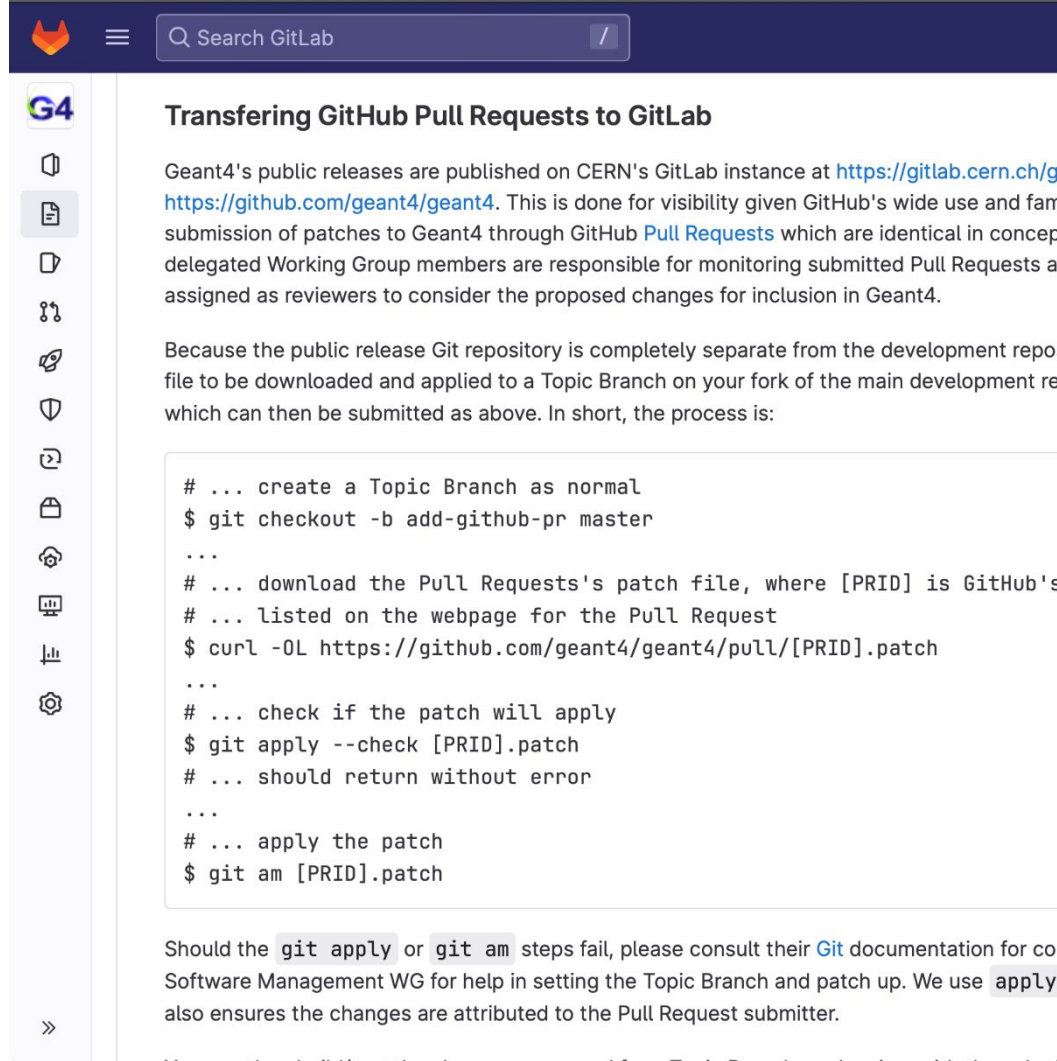
When assigned as reviewer, please take the time to **check the changes, suggesting update and approving** when finished!

Clarification: Rebasing for in-progress Merge Requests

- Have had reports in some MRs about problems rebasing Topic Branches
 - *We're here to help with this so **please don't hesitate to ask** but do let us know what errors/messages/problems you're seeing!*
 - *The main CONTRIBUTING.rst file also has information and links to good tutorials on the topic, let us know if this isn't clear or needs more info*
- What you **should not** do except in extrema is close the Merge Request and open a new one!
 - *Existing comments/reviews/commit histories are lost*
 - *Builds/CI are duplicated and waste resource*
- You also **do not have to rebase the MR** after it is submitted **unless** it conflicts with the current master or other in-progress MRs
 - *Rebasing does lose some review/comment info so you only want to do this in this exceptional circumstance*
 - *As with many git-related items, there is no single solution/process that accounts for every use case*

GitHub Contributions

- Mirror of public release repo
 - <https://github.com/geant4/geant4>
- GitHub more visible and do get occasional contributions from users as Pull Requests
- Integrating these whilst **keeping attribution** is straightforward, and process has initial documentation in CONTRIBUTING.rst



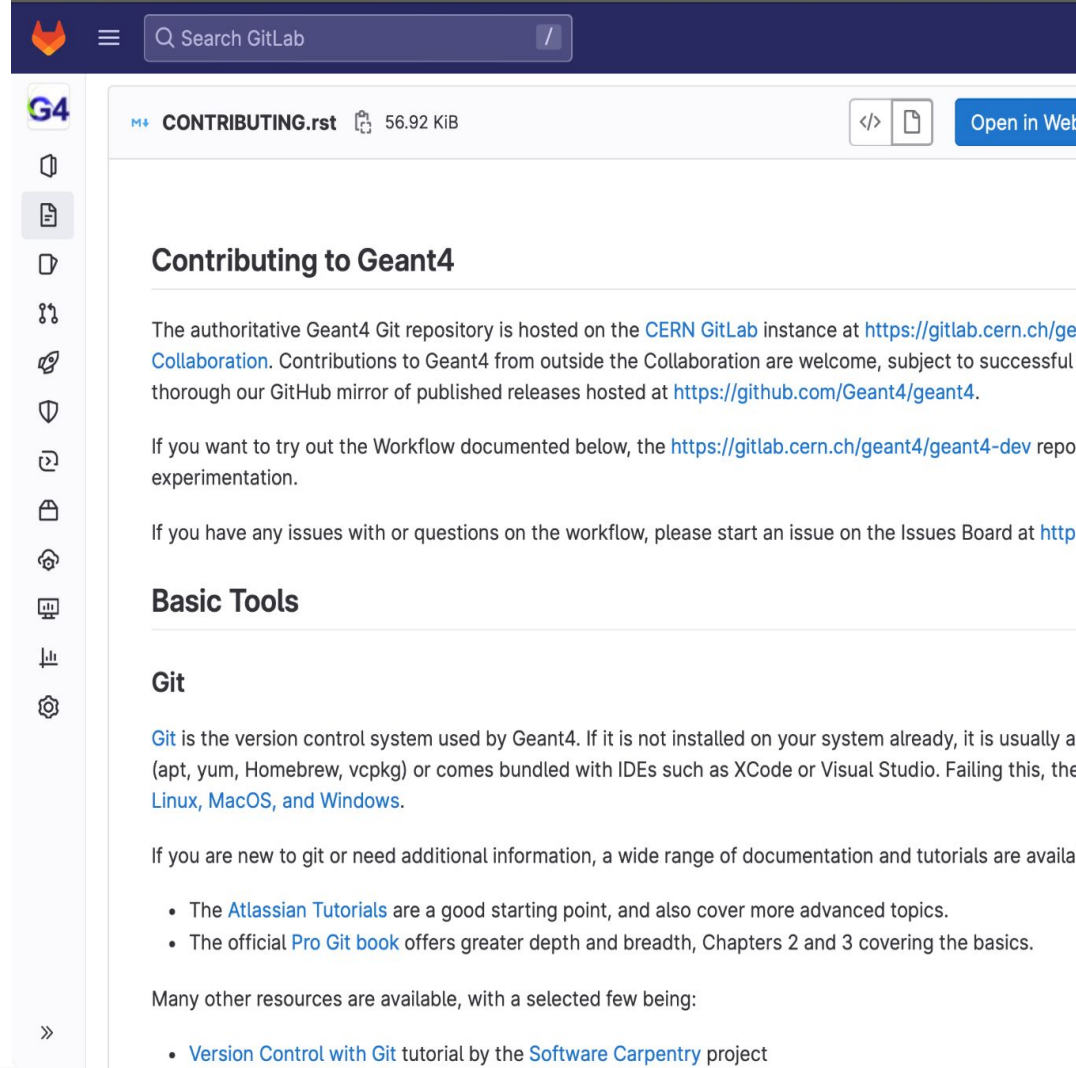
The screenshot shows a GitLab interface. At the top, there is a search bar with the text "Search GitLab". Below the search bar is a sidebar with various icons, including a GitLab logo, a home icon, a document icon, a folder icon, a link icon, a shield icon, a refresh icon, a lock icon, a house icon, a monitor icon, a bar chart icon, and a gear icon. The main content area is titled "Transferring GitHub Pull Requests to GitLab". It contains two paragraphs of text and a code block. The first paragraph explains that Geant4's public releases are published on CERN's GitLab instance and that GitHub pull requests are used for visibility. The second paragraph explains that the public release Git repository is completely separate from the development repository and that the process involves creating a topic branch, downloading the patch file, and applying it. The code block contains the following instructions:

```
# ... create a Topic Branch as normal
$ git checkout -b add-github-pr master
...
# ... download the Pull Requests's patch file, where [PRID] is GitHub's
# ... listed on the webpage for the Pull Request
$ curl -OL https://github.com/geant4/geant4/pull/[PRID].patch
...
# ... check if the patch will apply
$ git apply --check [PRID].patch
# ... should return without error
...
# ... apply the patch
$ git am [PRID].patch
```

Below the code block, there is a paragraph of text that says: "Should the `git apply` or `git am` steps fail, please consult their [Git](#) documentation for core Software Management WG for help in setting the Topic Branch and patch up. We use `apply` also ensures the changes are attributed to the Pull Request submitter."

GitLab Tutorial?

- Long term request to have a tutorial/seminar on GitLab
 - See Lorenzo's presentation later in relation to this
- To help develop this, **need your input** on what you want covered, e.g.
 - Just a "How to Merge Requests"?
 - More technical info, e.g. how Nightlies/Testing etc work?
- **Would aim to hold this in Jan/Feb 2024, but need the info above to prepare topics appropriately for your needs!**



The screenshot shows the GitLab web interface. At the top, there is a search bar with the text 'Search GitLab'. Below the search bar, the file name 'CONTRIBUTING.rst' is displayed with a file icon and the size '56.92 KIB'. On the right side, there are icons for code view and a button labeled 'Open in Web'. The main content area is titled 'Contributing to Geant4' and contains the following text:

The authoritative Geant4 Git repository is hosted on the [CERN GitLab](https://gitlab.cern.ch/geant4) instance at <https://gitlab.cern.ch/geant4> [Collaboration](#). Contributions to Geant4 from outside the Collaboration are welcome, subject to successful thorough our GitHub mirror of published releases hosted at <https://github.com/Geant4/geant4>.

If you want to try out the Workflow documented below, the <https://gitlab.cern.ch/geant4/geant4-dev> repository is used for experimentation.

If you have any issues with or questions on the workflow, please start an issue on the Issues Board at [https://gitlab.cern.ch/geant4/issues](#).

Basic Tools

Git

[Git](#) is the version control system used by Geant4. If it is not installed on your system already, it is usually available through package managers (apt, yum, Homebrew, vcpkg) or comes bundled with IDEs such as XCode or Visual Studio. Failing this, the [official Git installation instructions](#) cover [Linux](#), [MacOS](#), and [Windows](#).

If you are new to git or need additional information, a wide range of documentation and tutorials are available:

- The [Atlassian Tutorials](#) are a good starting point, and also cover more advanced topics.
- The official [Pro Git book](#) offers greater depth and breadth, Chapters 2 and 3 covering the basics.

Many other resources are available, with a selected few being:

- [Version Control with Git](#) tutorial by the [Software Carpentry](#) project

Build/Test/Package Topics



coverity.cern.ch

Geant4

Help Benjamin Morgan Enter CID(s)

Issues: By Snapshot | Outstanding Issues Filters: Issue Kind, Classification

CID	Type	Impact	Status	First...	Owner	Classifier	Severity	Action	Component	Category	File	Function
104900	Derefer...	Medium	New	08/29/23	ihrivnac	Unclassified	Unspecified	Undecided	Other	Null poin...	/build/gc...	WriteExtra
104899	Structur...	Medium	Triaged	08/29/23	gcosmo	Bug	Unspecified	Fix Submitt	Other	Control fl...	/build/gc...	PostStepDo
104898	AUTO_C...	Low	Triaged	08/29/23	ngoc	Bug	Minor	Fix Submitt	Other	Perform...	/build/gc...	InitializeNum
104897	Logically...	Medium	New	08/29/23	vnivanch	Unclassified	Unspecified	Undecided	Other	Control fl...	/build/gc...	ApplyYourse
104896	Uncheck...	Medium	New	08/29/23	vnivanch	Unclassified	Unspecified	Undecided	Other	Null poin...	/build/gc...	CalculateCh
104895	Derefer...	Medium	New	08/29/23	vnivanch	Unclassified	Unspecified	Undecided	Other	Null poin...	/build/gc...	Decaylt
104894	Unused ...	Medium	Triaged	08/29/23	gcosmo	Bug	Unspecified	Fix Submitt	Other	Incorrect...	/build/gc...	SampleSec
104890	Out-of-b...	High	New	06/16/23	ianpet	Unclassified	Unspecified	Undecided	Other	Memory	/build/gc...	update_fie

1 of 113 issues selected

G4RootHnFileManager.icc

```

50
51 // // create a new file
52 auto rfile = new tools::wroot::file(G4cout, fileName);
53 rfile->add_zipper('Z', toolx::compress_buffer);
54 rfile->set_compression(fFileManager->GetCompressionLevel());
55 if (rfile == nullptr) return false;
56
57 // no directory supported in this mode
58 auto hdirectory = &(rfile->dir());
59 if (hdirectory == nullptr) return false;
60

```

deref_ptr_in_call: Dereferencing pointer rfile, ["show details"]

CID 104900:(#1 of 5):Dereference before null check (REVERSE_NULL) ["select issue"]

CID 104900:(#2 of 5):Dereference before null check (REVERSE_NULL) ["select issue"]

CID 104900:(#3 of 5):Dereference before null check (REVERSE_NULL) ["select issue"]

CID 104900:(#4 of 5):Dereference before null check (REVERSE_NULL) ["select issue"]

CID 104900:(#5 of 5): Dereference before null check (REVERSE_NULL)
check_after_deref: Null-checking rfile suggests that it may be null, but it has already been dereferenced on all paths leading to the check.

104900 **Dereference before null check**

There may be a null pointer dereference, or else the comparison against null is unnecessary.

In G4RootHnFileManager<tools::histo::p1d>::WriteExtra(tools::histo::p1d *, G4String const &, G4String const &): All paths that lead to this null pointer comparison already dereference the pointer earlier (CWE-476)

eLearning: [Learn more about CWE-476](#) ... More

Triage

Classification:

Severity:

Action:

Ext. Reference:

Owner:

Enter comments (See the Triage History section below for previous comments)

Apply + Next Apply

Projects & Streams

Detection History

Triage History

Occurrences

Coverity Analysis

Good progress on addressing issues, but please keep this up as still **113 outstanding!**

Bugzilla/Geant4 – Problem List

[Home](#) | [New](#) | [Browse](#) | [Search](#) | [\[?\]](#) | [Reports](#) | [Help](#) | [New Account](#) | [Log In](#) | [Forgot Password](#)

Fri Sep 8 2023 11:04:54 CEST

[Hide Search Description](#)
Status: UNCONFIRMED, NEW, ASSIGNED

67 problems found.

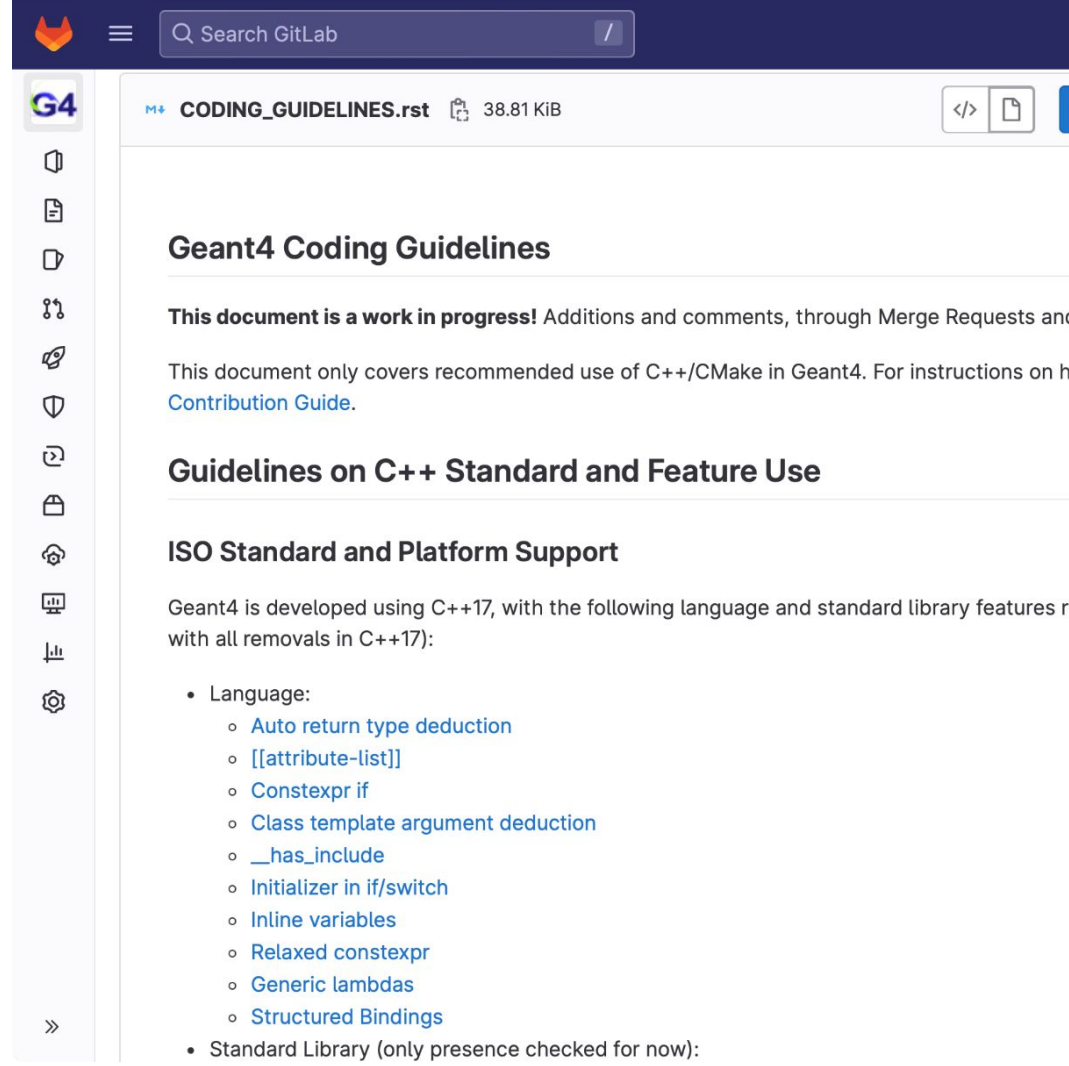
ID ▼	Product	Comp	Assignee	Status	Resolution	Summary	Changed
2563	Examples	purging_	John.Apostolakis	NEW	---	enabling and/or disabling certain physical processes makes the simulation stop after a few events	Thu 10:18
2561	Geant4	geometry	Pedro.Arce	NEW	---	Version 11.1.2 G4PhantomParameterisation producing nans in sensitive detector when running setregularstructureID(1).	Wed 08:51
2560	Geant4	processe	Vladimir.Ivantchenko	NEW	---	Program crashes halfway when simulating a large number of particles if use any reference physics list involving HP	2023-08-26
2559	Geant4	physics_	Vladimir.Ivantchenko	NEW	---	QBBC physics constructor and HadronPhysicsElasticHP	2023-08-23
2558	Geant4	physics_	Vladimir.Ivantchenko	ASSI	---	Unexpected Dependency of Neutron Energy Deposition with QBBC Physics List in Geant4 11.1.2 and latest master HEAD	2023-08-21
2557	Geant4	processe	Makoto.Asai	ASSI	---	Mesh treated as geometry boundary	Tue 11:11
2556	Geant4	cmake	Ben.Morgan	ASSI	---	Changes to FindEXPAT in CMake 3.27 break Geant4PackageCache.cmake file	2023-08-24
2554	Geant4	visualiz	laurent.garnier	ASSI	---	`G4OpenGLStoredQtViewer` assertion fails on debug build	2023-07-28
2552	Geant4	processe	loic.thulliez	ASSI	---	New neutron inelastic thermal scattering data for: h_water contains wrong cos value	2023-08-22
2551	Geant4	physics_	Vladimir.Ivantchenko	ASSI	---	An unphysical peak in the momentum distribution of pi- after interacting with thin slices of	2023-08-22

Bugzilla Reports

Important that reports are responded to in a timely manner, even if only to say “on backlog” 12

Coding Guidelines

- Draft/grab bag document on Geant4 development
 - *Guidelines on C++ Use*
 - *Coding style/clang-format*
 - *Use of clang-tidy, sanitizers*
 - *How to organise code and integrate in Geant4 libraries (technical details underlying Issue 122)*
- Is there interest in a tutorial/seminar on these?
 - *If so, what would you want covered?*
- How best to organise/house material? GitLab? Wiki? Web?



The screenshot shows a GitLab interface for a file named 'CODING_GUIDELINES.rst' (38.81 KiB). The file content is as follows:

Geant4 Coding Guidelines

This document is a work in progress! Additions and comments, through Merge Requests and

This document only covers recommended use of C++/CMake in Geant4. For instructions on how to contribute, see the [Contribution Guide](#).

Guidelines on C++ Standard and Feature Use

ISO Standard and Platform Support

Geant4 is developed using C++17, with the following language and standard library features recommended (with all removals in C++17):

- Language:
 - [Auto return type deduction](#)
 - [\[\[attribute-list\]\]](#)
 - [Constexpr if](#)
 - [Class template argument deduction](#)
 - [__has_include](#)
 - [Initializer in if/switch](#)
 - [Inline variables](#)
 - [Relaxed constexpr](#)
 - [Generic lambdas](#)
 - [Structured Bindings](#)
- Standard Library (only presence checked for now):

Issue 80: Official packages, CPack, DEB, RPM

- Official packages (also discussed in [Issue #80](#))
 - Available in [Arch Linux](#), [Conda](#), [Gentoo Linux](#), [Mac Ports](#), [NixOS](#), and [Spack](#)
- CMake build system can build binaries using CPack
 - Just run `cpack -G TYPE`, where TYPE is one of TGZ, DEB, RPM, etc
 - Needs appropriate config for dependencies to be added automatically
- Used [RPM Packaging Guide](#) to create starting point for official distribution
 - Created [SPEC file](#) with `geant4`, `geant4-{data,devel,examples}` packages
 - Published experimental RPM repo for CS8 built from the SPEC file at <http://lcgpackages.web.cern.ch/lcgpackages/test/geant4>
 - [Post in the Geant4 Forum](#) to let users try out the experimental packages
 - RPM packages can be used for building container images with Geant4
- [Merge Request](#) with rpm spec file and Dockerfiles in progress
- **Will need volunteers and effort to maintain and evolve**
 - Community effort as well, but good to have official packages for common platforms
 - See also [topic in Parallel 2A](#) on importance of this for Tutorials/Training



Issue 80: Geant4 Docker Images

- Original work by A. Dotti and W. Takase (similar efforts by J. Madsen)
- x86 and ARM images for Geant4 from C. Mancini:
 - <https://hub.docker.com/r/carlomt/geant4> (Images)
 - <https://github.com/carlomt/docker-geant4> (Sources)
- Data separate to keep image size small, mounted into container at runtime
- Example of building an image for a user application on top of this:
 - <https://github.com/carlomt/docker-dicom-g4example>
- How best to use/continue this line of packaging?
 - *Could be useful for training ([see Parallel 2A](#)), testing/profiling*
 - *RPM/Deb/etc better as “official” binaries, allowing easier user customization of their Docker images*
- What about the [Geant4 Virtual Machine](#)?



Issue 122: Improving structure of Geant4 libraries

- Long term effort to improve number/content of Geant4 libraries to:
 - Increase **coherence** by **grouping**, e.g. “libG4kernel.so” rather than “libG4{global, event, ...}”
 - Increase **modularity**, decrease **variance**, by **splitting**, e.g.
 - “libG4EMproc.so”, “libG4Hadproc.so” rather than “libG4processes.so”
 - “libG4gdml.so” rather than “libG4persistency.so” may/not have GDML
- Category/location of source code will also be reviewed after libraries
 - E.g. p/ABCs in “kernel”, concrete in “implementation”
- **First pass review of structure and work items now in Issue 122**
 - Open to all for discussion and input, and help needed from Working Group Coordinators on understanding some of the couplings found
- **Further working session(s) this week on this for more technical discussion on these items (likely Parallel 4B/5B)**

Issue 137: Improve Test Code Organization

- **Move per-category test/ directories from source/ to tests/**
 - Clearer separation of source and test code
 - Identify obsolete/redundant tests, or those suitable for use in CI vs validation
 - Identify potential for use of unit testing using, e.g. Google Test
- Final arbiter of “correctness” the integration and validation tests, but unit tests provide an extra layer of defence or to add regression checks.
- **See also Collaboration Evolution recommendations in [Plenary 3](#)**

- **source/**
 - .. move testing code from here to ..
- **tests/**
 - **tools/**
 - .. as is
 - **integration/**
 - **test00/**
 - .. all current `testXY`
 - **examples/**
 - CMakeLists.txt
 - .. current `tests/ctests/CMakeLists.txt`
 - **source/**
 - **global/**
 - **management/**
 - .. source/global/management/test files
 - .. or under "integration" if appropriate

Towards Geant4 12.0



What, already?!

- Might be several cycles away, but as a Major Release provides the opportunity for significant enhancements, we should be thinking about what these could be, and recording what and how to implement, **now**
 - *Clarify and prioritize work items given <1yr timescale for integration*
 - *Identify subtasks/preparation that can be done in Minor Releases*
 - *Clear deprecation warnings in preceding Minor Release(s)*
 - *Identify tasks that are lacking FTE or expertise, “Help Needed”*
- Though Major Release point always decided in concert with stakeholders, number/scope of items is a useful metric for when Major Release may be beneficial for everyone (reduce technical debt, enhance capabilities).
- **Want here to highlight related topics in this weeks Plenaries/Parallels**
 - *What tools to use to help us record/plan?*
 - *Development and R&D as sources of Major Release work items.*

The screenshot shows the GitLab Issue Boards interface for the 'geant4' project. The breadcrumb path is 'geant4 > geant4-dev > Issue Boards'. The filter bar shows 'Next Major Release...' selected, with a filter for 'Milestone = %Next Major Release'. Controls include 'Show labels', 'Group by' (set to 'None'), 'Edit board', and 'Create list'. The 'Open' column contains three issue cards: 1) 'Investigate/Review use of "Unity" builds for Continuous/Nightly' (enhancement, #33); 2) 'Review Thread-Local Singletons in Geant4' (discussion, enhancement, tasking, #65); 3) 'Review classes in global/HEPNumerics' (enhancement, #92). The 'Closed' column is currently empty.

Can/should also use an Issues Board to record our workplan items each year!
Might also be used to record User Requirements instead of JIRA

Use GitLab Issue Board?

Tasks/SubTasks as Issues with Milestone, linked to MRs when implemented to. GitLab ↔ JIRA is also a possibility. Related to topics in [Parallel 4A](#) on internal tools.

Technical Ideas 1: General/Modernization

- **Issue 152**: Improve usage of integral types for counters/”is optional”
 - *Good example of task that can be done through Minor Releases (implementation) with preparation for Major (user interfaces)*
 - *Good example of modernization, e.g. “G4int -> std::optional<size_t>”*
- **Issue 65**: Review use of Thread-Local Singletons
 - *Connected to thread id, so not easy to move tasks between threads*
 - *See Plenary 5: [Sub-Event Parallelism](#), [Initialization in Parallel](#)*
- **Issue 140**: Review use of global state/singletons/cleanup
 - *Reduce unexpected/unobvious behaviour*
 - *Reduce coupling*
 - *R&D on GPU use may be great source of feedback/info applicable to pure CPU mode as well, so please see results from this work in [Parallel 6B](#)*
 - *Also related to above topics in Plenary 5*



Technical Ideas: Workflows and Data Structures

- GPU-targeted projects like AdePT/Celeritas/Orange have explored various workflows and data structures for CPU-GPU co-working, **especially efficient memory layout/access**
 - *These may also be of benefit to pure CPU code (as in G4HepEM), so worth exploring*
 - *Similarly, track-based workflows on GPU \leftrightarrow sub-event parallelism on CPU?*
- Toolkit uses **raw pointers** (new/delete) **extensively**, canonical use case being inheritance (ptr-to-base), some others - all valid, but...
 - *Sometimes just copy-pasted without a valid design reason*
 - *Ownership of new-d resource not always obvious, even confusing, especially when passing raw pointers around*
 - *Awkwardness in code with collections of raw pointers (const-ness/de-refing)*
 - *Start thinking about how we can simplify this using standard tools like `unique_ptr` and other structures/libraries (e.g. [GuidelineSupportLibrary](#) or ranges ([range-v3/C++20](#)))*
 - *AdePT/Celeritas/ORANGE projects, though targeted at GPU, may also provide some useful insight/ideas/codes for pure Geant4/CPU.*
- Please see presentations across [Plenary 5](#), [Parallel 2B](#) and [Parallel 6B](#)!

Questions, Discussion

- GitLab/Git tutorial on Merge Request/Review process?
 - *Let me know what content you want to see in this, e.g. walkthrough ↔ technical details*
- Remember to use **Reviews, Coverity and Bugzilla!**
 - *Important for effective development and avoiding technical debt*
- Several development topics on build, packaging underway, Parallel 4B, 2A
- **We should start thinking/sketching out key topics for the next major release and how to record/track them - many ideas on these in sessions this week**

