

## Introduction

This page contains information on how to migrate hadronics category code for the upcoming Geant4 version 10. For a longer description of the issued summarized here please refer to the presentation given at the January 23rd, 2013 Had WG meeting: <https://indico.cern.ch/conferenceDisplay.py?confId=226961>.

*This is not a detailed document on multi-threading or Geant4 MT, but only a very rough list of todos for this category!*

The Geant4 Multi Threading Task Force has a twiki.

## What needs to be done

In current design of MT only memory consuming classes are shared among threads (geometry and EM physics tables). This mean that, for hadronics module, each thread has its own instances of the relevant classes.

As a consequence special attention has to be put with `static` and `global` variables: for their nature these variables are shared among threads. If their content vary between events, a possible issue arise: different threads can change the content of these variables while another thread is still using the old value.

This problem can be solved transforming the variable to be thread-local-storage. This is done adding a special keyword at the variable declaration as in the following example:

```
class G4MyHadClass {
private:
    static G4double aVariable;
};
```

is transformed to:

```
class G4MyHadClass {
private:
    static G4ThreadLocal G4double aVariable;
};
```

The `G4ThreadLocal` keyword in Linux is a typedef to `__thread`. For more details on this subject please see the document Mini guide to G4MT for developers (this document will become obsolete when an improved version will be integrated in the developers user guide).

## Important note

**It should be noted that these changes are needed only if the variable has a mutable state, if it is to be used in "read-only" (for example it contains a constant data table) the modification is not needed!** The change to thread-local-storage does not come for free: first of all there is a small memory penalty since each thread has its own copy of the variable, second there is some CPU degradation since there is a additional code for each thread to access its own private copy. In the final release of Geant4 version 10 this technique should be used only when needed. The top priority is to have a code that works correctly in multi-threaded, to guarantee this, in the current development version all instances of `static` and `global` variables have been transformed automatically to be `G4ThreadLocal` independently if they are mutable or not.

## Current Status (January 2013) of Hadronics module: Automatic conversion

The tag `geant4-09-06-ref-02` is the first one to contain multi-threaded transformed code. Once again: to avoid problems with possible shared variables *all* instances of `static` and `global` variables used in the code have been automatically transformed to thread-local-storage. As discussed in the Important Note this is probably not needed in most of the cases, since in hadronics module the use of static variables is generally intended to contain invariants of the code (data tables, constants values). We need to switch back the changes of the code. **It is very important that this effort is coordinated within the Hadronic WG, the Testing team and the people responsible for multi-threading** in doubt contact Andrea Dotti. The file <https://twiki.cern.ch/twiki/pub/Geant4/HadronicsMTNotes/HadronicsClasses.txt> contains a list of all files modified by the automatic conversion procedure.

## What to do if you want to remove the multi-threaded changes

If you are absolutely sure that the static variables you use in your code should be shared by all threads you can remove the `G4ThreadLocal` keyword from your code. However before removing this keyword you should make a tag with the current status of the code.

You can also use a special tag we created when we did the automatic conversion to remove at once all `G4ThreadLocal` keywords. To do this you need to (for example starting from first mt-aware tag):

```
g4svn co geant4-09-06-ref-02
cd <your-category>
g4svn switch trunk
g4svn tag <aNewInternalTag>
g4svn merge geant4-09-06-refmt-01-seq
```

The last command will revert back the changes to the code just before the multi-threading modifications were applied (see [\[\[#ListOfTags\]\]\[here\]](#) for a complete list of tags in this global tag). **Very important: This procedure is valid for hadronics module, you cannot apply blindly this procedure for other modules, please contact Andrea Dotti in case of questions!** Note that if you do a clean-up by hand class-by-class you do not need anymore to do the `merge`.

## List of directories to analyze

This is a prioritized list of the directories and categories to be checkedd (sub-dirs of `source/processes/hadronic`):

| Directory/module      | Number of files modified | Responsible/Contact | Status | Notes  |
|-----------------------|--------------------------|---------------------|--------|--|
| models/cascade        | 76                       | Mike                |        | High priority. Internal tag <code>hadr-casc-V09-06-01</code>               |
| cross_section         | 33                       | Witek/Andrea        |        | High priority  |
| models/de_excitation  | 27                       | Vladimir I.         |        |  |
| models/inclxx         | 23                       | Davide              |        |  |
| models/binary_cascade | 29                       | Gunter              |        | both: <code>im_r_matrix</code> and <code>binary_cascade</code> directories |
| models/neutron_hp     | 23                       | Tatsumi             |        |  |
| models/lll_fission    | 11                       | Dennis              |        | Proposed tag <code>hadr-lllfis-V09-06-01</code>                            |

|                         |    |                    |   |   |
|-------------------------|----|--------------------|---|---|
| models/coherent_elastic | 5  | Vladimir I.        | ● | Proposed tag <code>hadr-cohe-V09-06-02</code>     |
| models/parton_string    | 4  | Vladimir U./Gunter | ● |   |
| models/pre_equilibrium  | 4  | Vladimir I.        | ● | Accepted tag <code>hadr-pre-V09-06-01</code>      |
| models/lend             | 4  | Tatsumi            |   |   |
| stopping                | 4  | Julia              | ● | Accepted tag <code>hadr-stopping-V09-06-02</code> |
| models/management       | 2  | Witek + ?          |   |   |
| models/qmd              | 2  | Tatsumi            |   |   |
| models/quasi_elastic    | 2  | Witek              |   |   |
| models/rpg              | 22 | Dennis             | ● | Deprecated, do not touch                          |
| models/high_energy      | 25 |                    | ● | Deprecated, leave it like this?                   |
| models/low_energy       | 10 |                    | ● | Deprecated, leave it like this?                   |
| management              | 2  | Witek + ?          |   |   |
| util                    | 5  |                    |   |   |
| models/util             | 6  |                    |   |   |

● = Will not touch, leave as after automatic transformation

● = Started

● = Completed

● = Issues found, to be discussed what to do

## List of tags contained in `geant4-09-06-refmt-01-seq`

The tag `geant4-09-06-refmt-01-seq` includes the following list of tags on top of `geant4-09-06-ref-01`:

```

source/geometry/solids/specific          geom-specific-V09-06-03
source/materials                        materials-V09-06-00
source/processes/CMakeLists.txt         processes-cmk-V09-06-02
source/processes/GNUMakefile            processes-gmk-V09-06-01
source/processes/electromagnetic/highenergy emhighenergy-V09-06-02
source/processes/electromagnetic/lowenergy emlowen-V09-06-06
source/processes/electromagnetic/standard emstand-V09-06-06
source/processes/electromagnetic/utills  emutils-V09-06-03
source/processes/hadronic/CMakeLists.txt hadr-cmk-V09-06-00
source/processes/hadronic/GNUMakefile    hadr-gmk-V09-06-01
source/processes/hadronic/cross_sections hadr-cross-V09-06-03
source/processes/hadronic/models/CMakeLists.txt had-mod-cmk-V09-06-02
source/processes/hadronic/models/GNUMakefile had-mod-gmk-V09-06-02
source/processes/hadronic/models/binary_cascade had-binary-V09-06-02
source/processes/hadronic/models/im_r_matrix hadr-im_r-V09-06-00
source/processes/hadronic/models/inclxx  hadr-inclxx-V09-06-01
source/processes/hadronic/models/isotope_production -ARCHIVED-
source/processes/hadronic/models/management hadr-modman-V09-06-02
source/processes/hadronic/models/neutron_hp hadr-hpn-V09-06-03
source/processes/hadronic/models/qmd     hadr-qmd-V09-06-01
source/processes/hadronic/models/radioactive_decay radioactive_decay-V09-06-01
source/tracking                          tracking-V09-06-01

examples/extended/analysis/N03Con       exampleN03Con-V09-06-00
examples/extended/electromagnetic/TestEm7 testem7-V09-06-03
examples/extended/exoticphysics/monopole monopole-V09-06-00
examples/extended/hadronic/Hadr03       exhadr03-V09-06-01
examples/extended/optical/OpNovice      OpNovice-V09-06-00
examples/extended/runAndEvent           exRunAndEvent-V09-06-00
examples/extended/runAndEvent/RE02     exampleRE02-V09-06-01
examples/novice                          exNovice-V09-06-03
examples/novice/N03                     exampleN03-V09-06-00

tests/ctests                            ctests-V09-06-06

```

tests/test30  
tests/test67  
tests/test68

test30-V09-06-03  
test67-V09-06-01  
test68-V09-06-00

## Q&A and common pitfalls

This section has been moved in a dedicated twiki page.

---

This topic: Geant4 > HadronicsMTNotes

Topic revision: r23 - 2015-02-03 - AndreaDotti



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)