

IAEAphsp example status

Miguel A. Cortés-Giraldo

*Dep. Atomic, Molecular and Nuclear Physics
Universidad de Sevilla (Spain)*

miancortes@us.es

29th Geant4 Collaboration Meeting


Four Points Sheraton Hotel
Aci Castello, Catania (Italy), October 8, 2024.



Background

➤ **IAEAphsp**: Standardized format to use phase-space files produced from different codes.

 <http://www-nds.iaea.org/phsp>

- International Advisory Committee (IAC)
 - R. Jeraj
 - I. Kawrakow
 - C.-M. Ma
 - D.W.O. Rogers
 - F. Sanchez-Doblado
 - J. Sempau
 - J. Seuntjens
 - J.V. Siebers
 - P. Andreo
- Mailing Lists
 - Send mail to all members of the IAC
 - Register to the IAEA PHSP mailing list
- Medical Portal
 - Atomic and nuclear data for medical applications
- IAEA NAPC/NDS
 - Nuclear Data Section
- IAEA NAHU/DMRP
 - 

Phase-space database for external beam radiotherapy

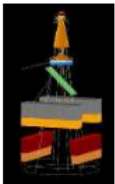
IAEA NAPC Nuclear Data Section
IAEA NAHU Dosimetry and Medical Radiation Physics Section

Project Officer: [Roberto Capote](#)

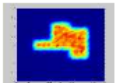
Objective: To build a database and disseminate representative [phase-space data](#) of accelerators and Co-60 units used in medical radiotherapy by compiling existing data that have been properly validated.

NEWS

[Dec 2009: Geant-4 interface to read/write the IAEA format released on December 14, 2009.](#)


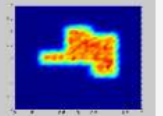


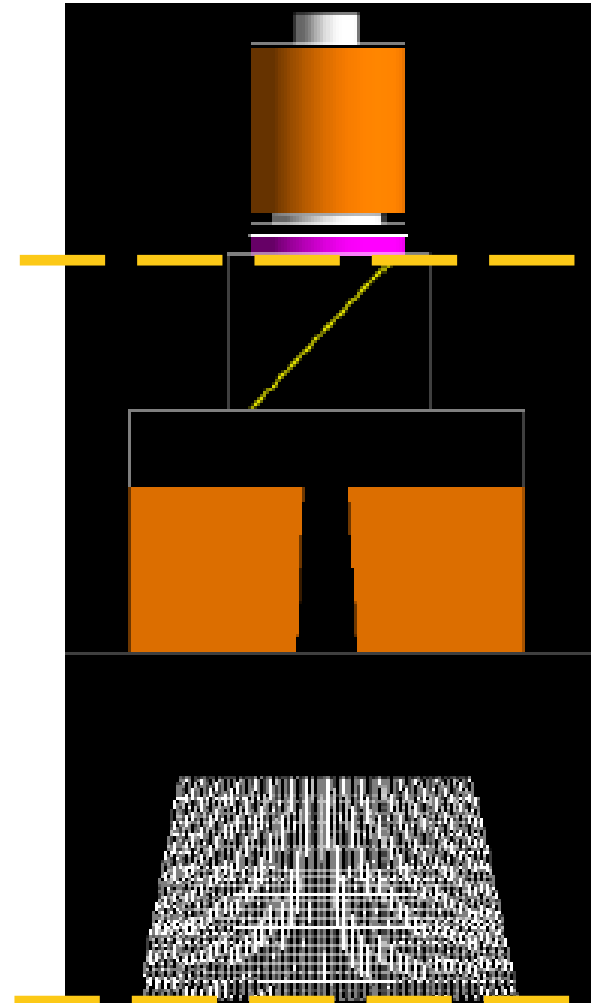
How to produce and submit phase-space data: The IAEA phsp format was designed to cover both phase-space files and event generators (see [phsp contents](#)). We have implemented the IAEA phsp format in a set of [read/write routines](#) (Updated: May 2011, see [readme file](#)). Native IAEA phsp format is available in EGSnrc and PENELOPE Monte Carlo codes. Geant4 interface to use the native IAEA phsp format is also [available](#). Once the validated phsp data is produced and documentation is published, [you may submit your phsp for review](#) using the [upload link here](#).



How to download phase-space data: You have to select a phsp data type among [Co-60 source](#), [linac electron](#) or [linac photon](#) phsps. For photon and electron PHSPs you may download the header first to decide which data you want to retrieve. Once decided you should download the PHSP data from the corresponding sub-directory. Please note that the first time access to the selected subdirectory could be slow.

Both the PHSP data and header should be present for the PHSP data to be accessible !

- Tech. Report IAEA-NDS-0484
 - 
- PHSP format
 - List of PHSP variables
- PHSP Header
 - How to fill header ...
- PHSP upload
 - Upload files
- PHSP to review
 - Files to review
- PHSP database
 - 
 - 1. Co-60 phsps
 - 2. Photon phsps
 - 3. Electron phsps



M.A. Cortés-Giraldo et al., IJRB 88: 200-8 (2012)

IAEAphsp code – current features

3

- **Focus:** to show **clearly** how to work with IAEAphsp files.
 - Very simple geometry & scoring (mostly to show how to include it in a concrete application)
- A controlled IAEAphsp file incorporated for testing purpose.
 - 1000 particles (200 of each kind: e-, e+, gamma, neutron, proton)
 - Controlled values of dynamic variables: energy, position, momentum, incremental history number (*n_stat*), etc.
- Compiled with latest version of IAEAphsp routines (sep-2013):

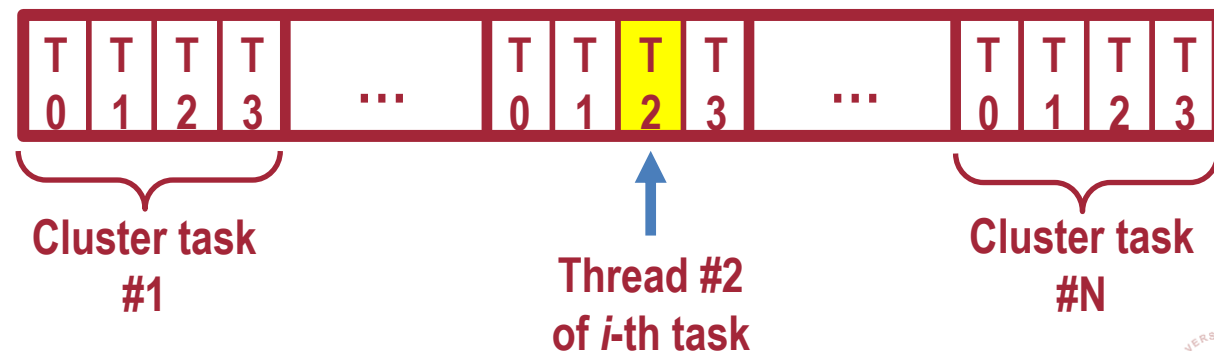
h iaea_config.h	
h iaea_header.h	C++ iaea_header.cc
h iaea_phsp.h	C++ iaea_phsp.cc
h iaea_record.h	C++ iaea_record.cc
h utilities.h	C++ utilities.cc
- **G4IAEAphspReader** class finished and tested.
 - **Messenger** class is also available.

G4IAEAphspReader MT class

- Derived from **G4VPrimaryGenerator**.
 - G4IAEAphspReader are **thread-local** objects.
- Object dynamically created in **ActionInitialization::Build()**
 - This was (at least for me!) the cleanest solution.
 - The file name has to be set prior object creation (no “empty” phsp reader is allowed).
- Thus, up to 30 threads are admitted.
 - As each **fstream** is associated to a `source_id` (either **reading** or **writing** a phsp file!), the limit may be lower when using a sources “slot” to write a phsp file.
 - (But see in next slide!)

- Each thread-local object defines a *source_id* defined internally in IAEA routines. (**MAX_NUM_SOURCES=30**)

- Support for cluster array-job of MT instances:
 - The fragment in file is limited accordingly.





IAEAphsp writer class(es)

5

- The writing operation is handled by thread-local and master objects.
- Each thread stores all relevant info in specific Run object.
 - Sort of *IAEAphsp stack*, not IAEA format yet.
- **After merging** all runs, particles stored are written into the IAEAphsp file.
- Various reasons:
 - No source_id is then needed simultaneously by reading and writing operations.
 - We never lose the possibility of using max source_id threads (currently set to 30).
 - If each thread created an IAEAphsp file, then these must be merged using external IAEA routines.

Work in progress...

test-writer ▾ IAEAphsp / + ▾ Compare History Find file Edit ▾ Code ▾

 **Split EndOfRunAction() into ClearRunVectors() and CloseOutputIAEAp...** ⋮ bab59b8d 
Miguel A. Cortes-Giraldo authored 14 hours ago

Name	Last commit	Last update
include	Split EndOfRunAction() into Cl...	14 hours ago
orig_writer	Add directory with the original...	21 hours ago
src	Split EndOfRunAction() into Cl...	14 hours ago
.README.txt	Initial commit	4 years ago
.gitignore	Add .gitignore	1 year ago
CMakeLists.txt	Update CMakeLists.txt to Gea...	1 year ago

- Testing of IAEAphsp writer
- UI commands for writer
- Code clean-up
- Documentation...



This work receives funding from Grant PID2021-123879OB-C21 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”, European Union.



@ miancortes@us.es

