11.9.7 Aperture ......................................................... 77
11.9.8 Compensator .................................................. 79
11.10 Patient Components ................................... 81
  11.10.1 Common Parameters .............................. 81
  11.10.2 Patient in DICOM Format ...................... 83
  11.10.3 Patient in XiO Format ............................. 84
  11.10.4 Imaging to Material Conversion ............. 84

12 Particle Sources ........................................... 89
  12.1 Introduction ............................................. 89
  12.1.1 Particle Names ..................................... 89
  12.2 Beam Sources ........................................... 90
  12.3 Emittance Sources ...................................... 92
  12.4 Isotropic Sources ........................................ 94
  12.5 Phase Space Sources ..................................... 95

13 Physics ......................................................... 101
  13.1 Introduction ............................................... 101
  13.2 Modular Physics Lists .................................. 101
    13.2.1 Physics Regions ................................... 102
    13.2.2 List of Available Modules ..................... 103
  13.3 Reference Physics Lists ............................... 104
  13.4 Optical Physics .......................................... 104
    13.4.1 Optical Photons ................................... 104
    13.4.2 Optical Surfaces .................................... 105
  13.5 Miscellaneous ........................................... 106
    13.5.1 User-Supplied Physics Lists .................. 106
    13.5.2 Multiple Physics Lists ......................... 106
    13.5.3 Production Thresholds ........................... 107
    13.5.4 Step Size ........................................... 107

14 Scoring ......................................................... 109
  14.1 Introduction ............................................... 109
  14.2 Volume Scorers ........................................... 109
    14.2.1 ProtonLET Scorer .................................. 110
  14.3 Surface Scorers .......................................... 111
  14.4 Phase Space Scorer ....................................... 112
    14.4.1 Phase Space Format ............................... 114
  14.5 Filtering Scorers ......................................... 116
  14.6 Output Specification ..................................... 119
  14.7 Miscellaneous ........................................... 122
    14.7.1 Binning in Dividable Components ............ 122
    14.7.2 Binning by Energy ................................ 122
    14.7.3 Binning by Time ................................... 122
    14.7.4 Splitting by Time Feature ..................... 123
    14.7.5 Statistical Information ......................... 124
    14.7.6 Change Component Color Based on Scoring .... 125
    14.7.7 Toggling a Scorer Off and On ................ 125
    14.7.8 Restoring Results from Files .................. 125

15 Graphics ......................................................... 127
# 16 Time Features

16.1 Linear, Sine, Cosine and Sqrt Functions ................................................................. 134
16.2 Step Function ........................................................................................................... 134
16.3 Combining Time Features for Complex Behaviors .................................................. 135

# 17 Variance Reduction

17.1 Introduction .............................................................................................................. 137
17.1.1 Specify the Split Geometry .................................................................................. 137
17.1.2 Define the Splitting Technique .......................................................................... 138
17.2 Geometrical Particle Splitting .................................................................................. 138
17.3 Importance Sampling .............................................................................................. 139
17.4 Weight Window ....................................................................................................... 140
17.5 Tracking Only Specific Particles ............................................................................ 141
17.6 Secondary Biasing .................................................................................................... 141

# 18 Basic

18.1 AllParameterForms.txt ........................................................................................... 143
18.2 DividedComponents.txt ........................................................................................ 146
18.3 Emittance_Gaussian.txt ........................................................................................ 148
18.4 Emittance_Twiss.txt ............................................................................................... 149
18.5 FlatteningFilter.txt ................................................................................................. 149
18.6 Isotope.txt ............................................................................................................... 150
18.7 LayeredMassGeometry.txt ..................................................................................... 151
18.8 OneBox.txt ............................................................................................................. 153
18.9 OneBoxRotate.txt ................................................................................................... 153
18.10 OneBoxTranslate.txt ............................................................................................. 153
18.11 PhysicsSetting.txt .................................................................................................. 154
18.12 ShapeTestWithAllParameters.txt ......................................................................... 155
18.13 ShapeTestWithOnlyRequiredParameters.txt ....................................................... 164
18.14 Spectrum.txt ......................................................................................................... 170
18.15 TwissPrimary.txt ................................................................................................... 171
18.16 TwoBeams.txt ...................................................................................................... 173

# 19 Graphics

19.1 ColorByOriginComponent.txt ............................................................................... 175
19.2 SolidSphere.txt ...................................................................................................... 177
19.3 TestGraphics.txt ..................................................................................................... 178
19.4 TwoProjections.txt ................................................................................................ 179

# 20 Scoring

20.1 ChargeInSphere.txt ............................................................................................... 181
20.2 Complex.txt ............................................................................................................ 182
20.3 Dose.txt .................................................................................................................. 185
20.4 DoseInVoxelMaterials.txt ....................................................................................... 186
20.5 DoseToMediumVsWater.txt .................................................................................... 187
20.6 DoseVolumeHistogram.txt ...................................................................................... 189
20.7 EnergyAndDose.txt ................................................................................................ 190
20.8 EnergyDivisions.txt ............................................................................................... 191
20.9 EnergyInBinnedCylinder.txt .................................................................................... 193
20.10 Filters.txt .............................................................................................................. 194
20.11 FoilToBox.txt ........................................................................................................ 206
20.12 FoilToCylinder.txt ................................................................................................. 208
20.13 FoilToSphere.txt .................................................................................................. 210
20.14 Gated.txt ............................................................................................................... 211
20.15 GeometryDivisions.txt ........................................................................................ 212
<table>
<thead>
<tr>
<th>Section</th>
<th>File Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.16</td>
<td>Histograms.txt</td>
<td>214</td>
</tr>
<tr>
<td>20.17</td>
<td>Inactive.txt</td>
<td>216</td>
</tr>
<tr>
<td>20.18</td>
<td>Ion.txt</td>
<td>216</td>
</tr>
<tr>
<td>20.19</td>
<td>Origin.txt</td>
<td>218</td>
</tr>
<tr>
<td>20.20</td>
<td>ParallelBoxRebinned.txt</td>
<td>219</td>
</tr>
<tr>
<td>20.21</td>
<td>SplitByTimeFeature.txt</td>
<td>221</td>
</tr>
<tr>
<td>20.22</td>
<td>Surfaces.txt</td>
<td>222</td>
</tr>
<tr>
<td>21</td>
<td>PhaseSpace</td>
<td>225</td>
</tr>
<tr>
<td>21.1</td>
<td>MultiRun_Write_ASCII.txt</td>
<td>225</td>
</tr>
<tr>
<td>21.2</td>
<td>ReadASCII.txt</td>
<td>226</td>
</tr>
<tr>
<td>21.3</td>
<td>ReadBinary.txt</td>
<td>227</td>
</tr>
<tr>
<td>21.4</td>
<td>ReadLimited.txt</td>
<td>228</td>
</tr>
<tr>
<td>21.5</td>
<td>WriteASCII.txt</td>
<td>229</td>
</tr>
<tr>
<td>21.6</td>
<td>WriteBinary.txt</td>
<td>230</td>
</tr>
<tr>
<td>21.7</td>
<td>WriteIonsASCII.txt</td>
<td>231</td>
</tr>
<tr>
<td>21.8</td>
<td>WriteLimited.txt</td>
<td>232</td>
</tr>
<tr>
<td>21.9</td>
<td>WriteROOT.txt</td>
<td>233</td>
</tr>
<tr>
<td>22</td>
<td>TimeFeature</td>
<td>237</td>
</tr>
<tr>
<td>22.1</td>
<td>BoxWithinBox.txt</td>
<td>237</td>
</tr>
<tr>
<td>22.2</td>
<td>CameraRotateAndZoom.txt</td>
<td>239</td>
</tr>
<tr>
<td>22.3</td>
<td>ChangingKEFilterByTimeFeature.txt</td>
<td>240</td>
</tr>
<tr>
<td>22.4</td>
<td>ColorChange.txt</td>
<td>241</td>
</tr>
<tr>
<td>22.5</td>
<td>CylinderGrowingInPhi.txt</td>
<td>242</td>
</tr>
<tr>
<td>22.6</td>
<td>Darkening.txt</td>
<td>244</td>
</tr>
<tr>
<td>22.7</td>
<td>Logo.txt</td>
<td>251</td>
</tr>
<tr>
<td>22.8</td>
<td>RotateResizeRecolor.txt</td>
<td>257</td>
</tr>
<tr>
<td>22.9</td>
<td>Rotation.txt</td>
<td>258</td>
</tr>
<tr>
<td>22.10</td>
<td>RunRandom_Mode.txt</td>
<td>261</td>
</tr>
<tr>
<td>22.11</td>
<td>RunSequential_Mode.txt</td>
<td>262</td>
</tr>
<tr>
<td>23</td>
<td>DICOM</td>
<td>265</td>
</tr>
<tr>
<td>23.1</td>
<td>DoseTo4DCT.txt</td>
<td>265</td>
</tr>
<tr>
<td>23.2</td>
<td>DoseToCT.txt</td>
<td>267</td>
</tr>
<tr>
<td>23.3</td>
<td>Implant.txt</td>
<td>268</td>
</tr>
<tr>
<td>23.4</td>
<td>ViewAbdomen.txt</td>
<td>270</td>
</tr>
<tr>
<td>24</td>
<td>Nozzle</td>
<td>273</td>
</tr>
<tr>
<td>24.1</td>
<td>RasterScanningPattern.txt</td>
<td>273</td>
</tr>
<tr>
<td>24.2</td>
<td>ScanningNozzle.txt</td>
<td>274</td>
</tr>
<tr>
<td>24.3</td>
<td>ScanningStationaryTarget.txt</td>
<td>284</td>
</tr>
<tr>
<td>24.4</td>
<td>ScanningTargetMovingHorizontal.txt</td>
<td>285</td>
</tr>
<tr>
<td>24.5</td>
<td>ScanningTargetMovingInDepth.txt</td>
<td>285</td>
</tr>
<tr>
<td>24.6</td>
<td>ScatteringNozzle.txt</td>
<td>286</td>
</tr>
<tr>
<td>24.7</td>
<td>ScatteringNozzle_run.txt</td>
<td>296</td>
</tr>
<tr>
<td>25</td>
<td>SpecialComponents</td>
<td>299</td>
</tr>
<tr>
<td>25.1</td>
<td>CAD.txt</td>
<td>299</td>
</tr>
<tr>
<td>25.2</td>
<td>DipoleMagnet.txt</td>
<td>300</td>
</tr>
<tr>
<td>25.3</td>
<td>MultiLeafCollimator.txt</td>
<td>301</td>
</tr>
<tr>
<td>25.4</td>
<td>MultiLeafCollimator_sequence.txt</td>
<td>303</td>
</tr>
<tr>
<td>25.5</td>
<td>MultiWire_Chamber.txt</td>
<td>304</td>
</tr>
<tr>
<td>25.6</td>
<td>Propeller.txt</td>
<td>305</td>
</tr>
<tr>
<td>25.7</td>
<td>Propeller_ContinuousRotation.txt</td>
<td>306</td>
</tr>
</tbody>
</table>
Note: A PDF version of the documentation is found by clicking the “Read the Docs” panel in the bottom-left corner of the website.
TOPAS wraps and extends the Geant4 Simulation Toolkit to provide an easier-to-use application for the medical physicist. TOPAS’s unique parameter control system lets you assemble and control a rich library of simulation objects (geometry components, particle sources, scorers, etc.) with no need to write C++ code.

Advanced users remain free to implement their own simulation objects in C++ code, and add them to TOPAS via an extension mechanism. While user-written objects benefit from underlying functionality of TOPAS base classes and the TOPAS parameter system, they can exploit the full flexibility of Geant4.

To discover the Geant4 version used by a specific version of TOPAS, please consult the History.

Note: Users should carefully read the Introduction to Parameter System, as well as the introductory parts of the sections Geometry Components, Particle Sources, Physics and Scoring. The rest of the documentation provides a detailed reference that you may just want to skim initially.
CHAPTER 2

Installation

The following README.txt is also bundled with the TOPAS software downloaded from toppasmc.org.

TOPAS Version 3.0.1

All use of TOPAS is governed by the TOPAS License Agreement provided in this directory as LICENSE.txt.

This README shows how to install and run TOPAS with and without user extensions.

In case of problems, see the TOPAS Monte Carlo User Forum at:
topasmc.org

These instructions are written for single user installations. Systems administrators performing multi-user installations may need to adapt these recipes.

These instructions assume the user has basic familiarity with the use of paths, shells and environment variables on their chosen operating system. For users who have that familiarity, it should be obvious how to adapt these recipes to their own installations. For others who do not have that familiarity, we suggest you first read some general tutorials for paths, shells and environment variables.

This product includes software developed by Members of the Geant4 Collaboration (http://cern.ch/geant4) and GDCM (http://gdcm.sourceforge.net)

0) Pre-Requisites:

Mac OSX 10.9 or newer: Install XQuartz
To run TOPAS on OSX 10.9 or newer, you must also install the X11 application from: http://xquartz.macosforge.org
Users of previous OSX versions do not need to perform this step.

Debian Users: Install the following:
apt-get install libexpat1-dev
apt-get install libgl1-mesa-dev
apt-get install libglu1-mesa-dev
apt-get install libxt-dev
apt-get install xorg-dev
apt-get install build-essential

1) Install TOPAS:

Unpack the tar.gz file:
tar -zxvf topas_3_0_*.tar.gz
Mac: Move the result so that you have /Applications/topas
Linux: Move the result so that you have ~/topas

2) Install Data Files:
The only part of Geant4 that you need to install are the data files. You do not need to download or build any other part of Geant4 since the necessary Geant4 libraries and header files are already include and pre-built in TOPAS. If you are simply upgrading from a previous TOPAS version that uses the same Geant4 version, you will already have these data files and can skip this step.

Mac: Download Geant4 Data files from the command line using curl:
curl -O http://geant4.cern.ch/support/source/G4NDL.4.5.tar.gz
curl -O http://geant4.cern.ch/support/source/G4PhotonEvaporation.3.2.tar.gz
curl -O http://geant4.cern.ch/support/source/G4RadioactiveDecay.4.4.tar.gz
curl -O http://geant4.cern.ch/support/source/G4NEUTRONXS.1.4.tar.gz
curl -O http://geant4.cern.ch/support/source/G4PII.1.3.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4NEUTRONXS.1.4.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
curl -O http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz

Linux: Download Geant4 Data files from the command line using wget:
wget http://geant4.cern.ch/support/source/G4NDL.4.5.tar.gz
wget http://geant4.cern.ch/support/source/G4PhotonEvaporation.3.2.tar.gz
wget http://geant4.cern.ch/support/source/G4RadioactiveDecay.4.4.tar.gz
wget http://geant4.cern.ch/support/source/G4NEUTRONXS.1.4.tar.gz
wget http://geant4.cern.ch/support/source/G4PII.1.3.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz
wget http://geant4.cern.ch/support/source/G4SAIDDATA.1.1.tar.gz

Mac: Unpack and move to:
/Applications/G4Data/G4EMLOW6.48
/Applications/G4Data/G4NDL.4.5
/Applications/G4Data/PhotonEvaporation3.2
/Applications/G4Data/RadioactiveDecay4.4
/Applications/G4Data/G4SAIDDATA1.1
/Applications/G4Data/G4NEUTRONXS1.4
/Applications/G4Data/G4PII.1.3
/Applications/G4Data/RealSurface1.0
TOPAS Documentation, Release 3.0

/Linux: Unpack and move to:
~/G4Data/G4EMLOW6.48
~/G4Data/G4NDL4.5
~/G4Data/PhotonEvaporation3.2
~/G4Data/RadioactiveDecay4.4
~/G4Data/G4SAIDDATA1.1
~/G4Data/G4NEUTRONXS1.4
~/G4Data/G4PII1.3
~/G4Data/RealSurface1.0
~/G4Data/G4ABLA3.0
~/G4Data/G4ENSDFSTATE1.2.1
~/G4Data/G4TENDL1.0

3) Set up the environment:
Mac:
export G4LEDATA=/Applications/G4Data/G4EMLOW6.48
export G4NEUTRONHPDATA=/Applications/G4Data/G4NDL4.5
export G4LEVELGAMMADATA=/Applications/G4Data/PhotonEvaporation3.2
export G4RADIOACTIVEDATA=/Applications/G4Data/RadioactiveDecay4.4
export G4SAIDDATA=/Applications/G4Data/G4SAIDDATA1.1
export G4NEUTRONXSDATA=/Applications/G4Data/G4NEUTRONXS1.4
export G4PIIDATA=/Applications/G4Data/G4PII1.3
export G4REALSURFACEDATA=/Applications/G4Data/RealSurface1.0
export G4ABLADATA=/Applications/G4Data/G4ABLA3.0
export G4ENSDFSTATEDATA=/Applications/G4Data/G4ENSDFSTATE1.2.1
export G4TENDLADATA=/Applications/G4Data/G4TENDL1.0
export DYLD_LIBRARY_PATH=/Applications/topas/libexternal:$DYLD_LIBRARY_PATH

Linux Bourne shell:
export G4LEDATA=/G4Data/G4EMLOW6.48
export G4NEUTRONHPDATA=/G4Data/G4NDL4.5
export G4LEVELGAMMADATA=/G4Data/PhotonEvaporation3.2
export G4RADIOACTIVEDATA=/G4Data/RadioactiveDecay4.4
export G4SAIDDATA=/G4Data/G4SAIDDATA1.1
export G4NEUTRONXSDATA=/G4Data/G4NEUTRONXS1.4
export G4PIIDATA=/G4Data/G4PII1.3
export G4REALSURFACEDATA=/G4Data/RealSurface1.0
export G4ABLADATA=/G4Data/G4ABLA3.0
export G4ENSDFSTATEDATA=/G4Data/G4ENSDFSTATE1.2.1
export G4TENDLADATA=/G4Data/G4TENDL1.0
export LD_LIBRARY_PATH=/topas/libexternal:/$LD_LIBRARY_PATH

Linux C shell:
setenv G4LEDATA ~/.G4Data/G4EMLOW6.48
setenv G4NEUTRONHPDATA ~/.G4Data/G4NDL4.5
setenv G4LEVELGAMMADATA ~/.G4Data/PhotonEvaporation3.2
setenv G4RADIOACTIVEDATA ~/.G4Data/RadioactiveDecay4.4
setenv G4SAIDDATA ~/.G4Data/G4SAIDDATA1.1
setenv G4NEUTRONXSDATA ~/.G4Data/G4NEUTRONXS1.4
setenv G4PIIDATA ~/.G4Data/G4PII1.3
setenv G4REALSURFACEDATA ~/.G4Data/RealSurface1.0
setenv G4ABLADATA ~/.G4Data/G4ABLA3.0
setenv G4ENSDFSTATEDATA ~/.G4Data/G4ENSDFSTATE1.2.1
setenv G4TENDLADATA ~/.G4Data/G4TENDL1.0
setenv G4ENSDFSTATEDATA ~/.G4Data/G4ENSDFSTATE1.2.1
4) Run TOPAS:
Mac: cd /Applications/topas
Linux: cd ~/topas

To run a whole set of examples:
source rundemos.csh

To run a single example:
cd to the directory that the example is in, then run topas from there, for example:
cd examples/SpecialComponents
././topas MultiLeafCollimator_sequence.txt

To test TOPAS with DICOM:
Unzip the example DICOM directories in examples/DICOM
cd examples/DICOM
././topas ViewAbdomen.txt

Remember that some parameters have changed since Topas 2.0.
We provide a script that you can run to convert your parameter files from Topas2 style to Topas3 style.
Download the script:
Mac: curl -O www.slac.stanford.edu/~perl/topas2to3
Linux: wget www.slac.stanford.edu/~perl/topas2to3

Then for any OS:
chmod +x topas2to3
/path/to/topas2to3 your_parameter_file.txt

This will replace your_parameter_file.txt with a new copy, and will save the previous version with a .bak file extension.

5) To add User Extensions:
You will need a tool called CMake (version 3.3 or newer).
Type "which cmake" to see if you already have this tool and "cmake --version" to see what version you may have.

If you need to install CMake, install from a binary distribution at:
http://www.cmake.org/cmake/resources/software.html
Run the cmake.app and follow the instructions in CMake's menu item: "Tools"... "How to Install for Command Line Use"

Because our underlying Geant4 version requires a compiler that supports C++11, you will need gcc >=4.8.3
To check your gcc version, gcc --version
If your gcc version is not sufficient, ask your sysadmin if there is already a newer version in some non-default installation on your system. They may be able to tell you how to switch to that version.
Otherwise, you will need to install a newer version or ask your sysadmin to do so.
Then locate the libstdc++.so file and add this directory to your LD_LIBRARY_PATH environment variable
(`export LD_LIBRARY_PATH=/path/to/gcc/lib:$LD_LIBRARY_PATH`).
Unzip the full set of Geant4 header files from topas/Geant4Headers.zip. You should end up with a new directory: topas/Geant4Headers. Do not use headers from any other version of Geant4 as they could appear to run OK but give wrong results!

Place your extra TOPAS code into a directory that is NOT inside of the topas directory. For example, you might have TOPAS in:
/Applications/topas
and the extensions in:
/Applications/topas_extensions

Mac: cd /Applications/topas
Linux: cd ~/topas

```bash
cmake -DTOPAS_EXTENSIONS_DIR=/Applications/topas_extensions
make -j8
```

In the make statement, the value after `-j`, such as `-j8` indicates how many threads make should use at once. This depends on your computer's capabilities. A typical quad-core machine with hyperthreading run efficiently with 8 threads.

Don't worry about any "Warning" messages. Only worry about messages that say "Error". If you get any error messages, confirm that you unzipped the Geant4 header files as explained above. Also check that you have an appropriate compiler (see notes earlier in this section).

Then set up the environment and run as in steps 3 and 4 above.

```
CMake caches the name of your extensions directory and watches for subsequent changes there. If you make changes to any of the extensions you were already including (such as while you are debugging your extensions), you just need to re-run make -j8
```

If you add additional extensions to your extensions directory, again run the full `cmake -DTOPAS_EXTENSIONS_DIR=/Applications/topas_extensions` `make -j8`
TOPAS Software

- Joseph Perl
- Jungwook Shin
- Jan Schuemann
- Bruce Faddegon
- Harald Paganetti
- Jose Ramos
- Aimee McNamara
- David Hall
### TOPAS Documentation

<table>
<thead>
<tr>
<th>Website Creator</th>
<th>David Hall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lead Author</td>
<td>Joseph Perl</td>
</tr>
</tbody>
</table>
| Additional Authors    | • Jungwook Shin  
|                       | • Jan Schuemann  
|                       | • Jose Ramos  
|                       | • David Hall |
| Editors               | • Joseph Perl  
|                       | • David Hall |
| User Contributions    | • Aleksandra Biegun  
|                       | • Fada Guan |
If you use TOPAS, please be sure to cite the following paper (PubMed):


To discover the Geant4 version used by a specific version of TOPAS, please consult the History.

Here is a BibTeX entry to aid use within LaTeX:

```bibtex
@article{TOPAS,
    author = {Perl, J. and Shin, J. and Schumann, J. and Faddegon, B. and Paganetti, H.},
    title = {{TOPAS: An innovative proton Monte Carlo platform for research and clinical applications}},
    journal = {Medical Physics},
    pages = {6818},
    volume = {39},
    year = {2012},
    pmid = {23127075},
    doi = {10.1118/1.4758060}
}
```
CHAPTER 5

User Support and Contributions

User Support

Help can be requested from TOPAS developers and other users at the User Forum. Note that you must login as a registered user before you can access the forums.

User Contributions

- Feedback and feature requests are best communicated via the User Forum described above.
- If you would like to share your TOPAS extensions or parameter files with other users, please get in touch
- Contributions to this documentation website are also most welcome, but does require a GitHub account:
  - Click “Edit on GitHub” in the top-right corner of the page you wish to edit
  - This takes you to the page source at GitHub.com
  - Click the pencil icon to fork the repository and edit the page source
  - The source format is reStructuredText
  - After making your changes, open a pull request
TOPAS wraps and extends the Geant4 Simulation Toolkit. The following table lists the TOPAS versions that introduced a new version of the Geant4 library.

<table>
<thead>
<tr>
<th>TOPAS version</th>
<th>Geant4 version</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0.0</td>
<td>10.2.p01</td>
</tr>
<tr>
<td>2.0.0</td>
<td>10.1.p02</td>
</tr>
<tr>
<td>1.0.0</td>
<td>9.6.4</td>
</tr>
</tbody>
</table>

A detailed list of changes introduced by each TOPAS version can be found below:

### 3.0 Series

The main features introduced by the 3.0 series were:

- n-tuple output from custom scorers
- new extensions mechanism
- redesigned beam source parameter interface
- generalized imaging to material conversion
- electric fields
- faster graphics performance

We also updated the underlying Geant4 version to Geant4.10.2.p01.

We took the opportunity of a major release to revise some parameter names with the benefit or hindsight and looking towards future releases. Consequently, we provided a topas2to3 script to help in upgrading parameter files.

#### 3.0.1 (2016-06-02)

*Fixed: drawing to multiple OpenGL graphics views not working correctly* We have been struggling with some issues in OpenGL graphics since we moved to Geant4.10.02. In particular, there were issues if you drew more...
than one view at a time. We now believe this is fixed, with the exception of the Debian8 build. Behavior is highly dependent on the specific operating system, and we (and Geant4) are still working to get this just right.

**Fixed: compiler issues when adding extensions** Building with extensions should now work on all operating systems. It is no longer necessary to run cmake twice in a row, and our cmake does a better job now of setting the required C++11 flags. You will still need to update your compiler if your default compiler does not support C++11. This is now discussed in the last section of the README.

**Fixed: bug in filter extensions** Can now add multiple filter extensions simultaneously.

**Improved: geometry component types now case-insensitive** It no longer matters what case you use in `Ge/*/Type` values. So, for example, you could use "tsBox", "TsBox" or "tsbox". This also applies to user-supplied extension components.

### 3.0.0 (2016-06-01)

**Added: n-tuple output from custom scorers** If you’re not familiar with n-tuples, you may be interested to know that you’ve already been using a version of one if you were outputting particle information to phase space.

Our new design gives you very fine control over what information your scorer will write.

In your scorer’s constructor you define each column and its data type. For floating point columns you also specify the unit string, similar to how the `SetUnit()` method is used in other scorers. In the `ProcessHits()` method, you then compute the variables you want to output and then call `fNtuple->Fill()`. Buffered file writing is automatically handled. Output can be to ASCII, binary and ROOT data formats.

**Added: phase space scorers can be output to ROOT format** This is a binary data format associated with the ROOT data analysis framework.

**Improved: overhaul of extension mechanism** No more editing the CMakeLists file or adding a clause into the TsExtensionManager.cc.

Just create a folder anywhere on your computer, put your extensions into it (you can even organize them by subfolders within this folder), add one special comment line to the top of each cc file, and run CMake, telling it where your folder is with the `-DTOPAS_EXTENSIONS_DIR` option. Our CMake scripts will then automatically interweave your code in with ours, and all will work. Please note that, with the introduction of n-tuple scorers, existing scorers will require some additional modifications to distinguish them as “binned scorers”.

**Improved: revised parameter interface for beam and isotropic sources** You can now provide a cutoff (like a collimation) to the angular spread distribution. This improvement was sparked by a request from Christian Sommer.

The overall parameter set has been revised to be, we think, easier to understand and remember.

- **Type** = {Beam, Phasespace, Emittance, Isotropic}
  - **BeamPositionDistribution** = {None, Flat, Gaussian} (if Beam)
    - BeamPositionCutoffShape = {Rectangle, Ellipse} (if Flat/Gaussian)
    - BeamPositionCutoffX/Y (if Flat/Gaussian)
    - BeamPositionSpreadX/Y (if Gaussian)
  - **BeamAngularDistribution** = {None, Flat, Gaussian} (if Beam)
    - BeamAngularCutoffX/Y (if Flat/Gaussian)
    - BeamAngularSpreadX/Y (if Gaussian)

The previous option `BeamShape = “Point”` is now chosen via `BeamPositionDistribution = "None"`, and the previous option `BeamShape = “Isotropic”` is now chosen via `Type = “Isotropic”`. 
Here is the list of parameters that already existed, but now have new names:

- BeamXYDistribution → BeamPositionDistribution
- BeamShape → BeamPositionCutoffShape
- BeamHWX → BeamPositionCutoffX
- BeamHWY → BeamPositionCutoffY
- BeamStandardDeviationX → BeamPositionSpreadX
- BeamStandardDeviationY → BeamPositionSpreadY

**Improved: default source removed** The default particle source parameters, So/Default/*, have been removed. While revising particle sources, we changed our mind about whether we should provide any default particle source parameters. We see so many differences in what different users need, that we now believe it is important that every user carefully consider all of their particle source parameters. Accordingly, we removed the default parameters So/Default/*. All examples now include a full specification of their particle source parameters.

We still bundle a built-in source, but we have renamed it So/Demo to emphasize that it should be used only for demonstration and teaching purposes. Users who write their own parameter files should fully specify their Particle Source.

**Improved: changed the default water material** We removed our custom Water (mean excitation energy 75 eV) in favor of G4_WATER (78 eV).

For some time, we had been using a custom material, Water, with a mean excitation energy of 75 eV, instead of Geant4’s pre-defined G4_WATER, which has a mean excitation energy of 78 eV. With newer Geant4 physics, we find we get better agreement at MGH when we use 78 eV. Accordingly, we have removed our material Water and switched all of our examples to G4_WATER.

We created a new material named Water_75eV in case you have calibrated with this material and really want to stick with that.

**Added: support for additional imaging to material conversion** Support Dual Energy CT, Multi-Energy CT and other complex ways of assigning material. We have generalized our imaging to material conversion. You can now provide your own extension class to control how TOPAS assigns materials in the patient. Input can come from one or more image files and the imaging modality is not limited to CT. TOPAS pulls the information out of the images for you, and gives your class the one or more values per voxel. Your class then determines what material to then assign for this voxel.

New parameters are:

```plaintext
i:Ge/Patient/NumberOfEnergies = 1 # defaults to 1
sv:Ge/Patient/DicomModalityTags = 1 "CT" # defaults to just CT
s:Ge/Patient/ImagingToMaterialConverter = "Schneider"
```

You will note that we renamed the parameter HUtoMaterialConversionMethod to ImagingToMaterialConverter to emphasize that input need not necessarily be Hounsfield Units.

**Added: field handling extended to include electric fields** Where before we had:

```plaintext
Ge/MyComponent/MagneticField = "Dipole" # or "Quadrupole", "Map"
```

We now have:

```plaintext
Ge/MyComponent/Field = "DipoleMagnet" # or "QuadrupoleMagnet", "MappedMagnet",
                   "UniformElectroMagnetic"
```
The latter can have both magnetic and electric components. To specify a pure electric field, use “UniformElectroMagnetic” while specifying a zero MagneticFieldStrength. You can also write your own extension field class that provides any other Electric, Magnetic or ElectroMagnetic field.

To make way for ElectricField parameters, we renamed some parameters:

- DirectionX -> MagneticFieldDirectionX
- DirectionY -> MagneticFieldDirectionY
- DirectionZ -> MagneticFieldDirectionZ
- Strength -> MagneticFieldStrength
- GradientX -> MagneticFieldGradientX
- GradientY -> MagneticFieldGradientY
- 3DTable -> MagneticField3DTable
- Stepper -> FieldStepper
- StepMinimum -> FieldStepMinimum
- DeltaChord -> FieldDeltaChord

**Improved:** magnetic field setup from 3D field maps  The new code does a much better job of handling various field maps. Thanks to Eric Able of Varian Medical Systems for his extensive prototype work.

**Added:** support for variable density materials  Driven by needs from imaging to material conversion, we have added a way that you can easily define a set of materials that differ only in density:

```plaintext
Ma/MyMaterial/VariableDensityBins = 100
Ma/MyMaterial/VariableDensityMin = .1
Ma/MyMaterial/VariableDensityMax = 10.
```

will generate 100 versions of MyMaterial, with densities varying from .1 x normal to 10. x normal.

The material names will then be like:

```
MyMaterial_VariableDensityBin_0
MyMaterial_VariableDensityBin_1
...
MyMaterial_VariableDensityBin_99
```

Note that numbering starts at zero.

**Improved:** support for multiple slice thicknesses in TsDicomPatient  This capability was previously restricted to the TsXioPatient.

**Improved:** OpenGL graphics are dramatically faster  Especially true for patient geometry. If you were avoiding displaying patient geometry, it’s time to try it again.

**Added:** new export options for OpenGL Graphics  Where before we had:

```
Gr/MyView/CopyOpenGLToEPS
```

we now have that plus three new options:

```
Gr/MyView/CopyOpenGLToPDF
Gr/MyView/CopyOpenGLToSVG
Gr/MyView/CopyOpenGLToPS
```
Added: new syntax for specifying vectors  New parameter expressions let you set vector parameters from other vector parameters:

\[
\begin{align*}
dv &= \text{name of double vector parameter} \text{ unit} \\
dv &= \text{number} \times \text{name of double vector parameter} \text{ unit} \\
dv &= \text{name of unitless or integer parameter} \times \text{name of double vector parameter} \text{ unit} \\
\text{uv} &= \text{name of unitless vector parameter} \\
\text{uv} &= \text{number} \times \text{name of unitless vector parameter} \\
\text{uv} &= \text{name of unitless or integer parameter} \times \text{name of unitless vector parameter} \\
\text{iv} &= \text{name of integer vector parameter} \\
\text{iv} &= \text{integer} \times \text{name of integer vector parameter} \\
\text{iv} &= \text{name of integer parameter} \times \text{name of integer vector parameter} \\
\text{bv} &= \text{name of boolean vector parameter} \\
\text{sv} &= \text{name of string vector parameter}
\end{align*}
\]

This makes it easier to adjust existing vector parameters in file hierarchies.

Improved: G4Box, G4Tubs and G4Sphere components removed  You should instead use TsBox, TsCylinder and TsSphere. These provide all the same functionality, but also support divisions. We have done this both to simplify the underlying TOPAS code (simpler means less likelihood for bugs), and because we have seen many cases where someone tried to apply divisions to G4Box, G4Tubs or G4Sphere, and had a hard time figuring out why this wasn’t working (this has even happened to us during live demos).

Added: specify material per voxel for any divided component  This means you can create complex phantoms directly from the parameter system:

\[
\text{sv/Ge/Phantom/VoxelMaterials} = 100 "\text{G4\_WATER}" "\text{G4\_WATER}" "\text{Air}" "\text{Air}" "\text{G4\_WATER}" ...
\]

Works for all three kinds of divided components: TsBox, TsCylinder and TsSphere.

Improved: TsBox and TsCylinder allow parameterizations in parallel worlds  The underlying limitation that was preventing this has been resolved in the new Geant4 version. The restriction against using parameterization within a parallel world now only applies to TsSphere (as the underlying issue for this case is still present in Geant4).

Improved: support larger numbers of histories  Various Counters have been changed from int to long to accommodate larger numbers of histories. With the move to multi-threading, we now have users running so many histories in a single session that various counters exceeded the size of our internal counters. There remain some limits within Geant4 itself, so we enforce a maximum of \(10^9\) histories per run.

We also found a way to allow you to have more than \(10^9\) histories in a single TOPAS session. The solution is to break these histories into multiple Geant4 runs. Originally, the parameters Tf/TimeLineStart, Tf/TimeLineEnd and Tf/NumberOfSequentialTimes were intended to let you have different runs at different times (TOPAS Time Features). But if you leave TimeLineEnd the same as TimeLineStart (and by default they are both 0), and just set Tf/NumberOfSequentialTimes to some value greater than 1, you will have multiple runs, and each can have up to \(10^9\) histories, but the total can be much larger.

Improved: ProtonLET scorer extended to very low density materials  Our current LET scorer gives values that are too high in air, where the mean path length between discrete processes can be larger than the voxel size. This can be avoided by neglecting secondary electrons, so we introduce the \(\text{NeglectSecondariesBelowDensity}\) parameter, whose default value is 0.1 g/cm³.

Even when you do this, rare events that produce very low energy protons (e.g. a recoiling hydrogen nucleus) will produce spikes in LET. This is also seen in the PreStepLookup version of the scorer. They are not seen in the fluence-averaged version of the scorer, since they are rare events. For this reason we introduce a \(\text{UseFluenceWeightedBelowDensity}\) parameter, whose default value is zero. We disable this by default because it is strange to mix both types of LET in a single distribution, and could be significantly wrong at the
end of range. We expect users to want to enable this when making a pretty plot of LET to overlay on a CT scan, without spikes in cavities and outside the patient.

**Improved: convenience method GetIndex for custom scorers** Scorers can now easily obtain the voxel indices from hits in divided or parameterized components. The base class TsVScorer now provides a convenience method, G4int GetIndex(G4Step*). This is convenient for some expert users and also hides the GetIndex method that we don’t want people trying to use from the G4VPrimitiveScorer (since the latter doesn’t perform as the user would expect).

**Improved: DoseToWater and DoseToWaterBinned scorers are unified** DoseToWaterBinned was a way of scoring dose to water that improved speed at some cost to accuracy by pre-calculating stopping power ratios. We now offer only one scorer, DoseToWater. To get the previous behavior of DoseToWaterBinned, add the optional parameter:

```
:b:Sc/MyScorer/PreCalculateStoppingPowerRatios = "True" # defaults to "False"
```

The same parameter is also available for the DoseToMaterial scorer.

**Improved: removed our custom EM Physics Module from our Default Physics List** In TOPAS 2.0 we provided a custom EM physics module, tsem-standard_opt3_WVI, that attempted to use the new WentzelVI model of multiple Coulomb scattering (MCS). Now that this MCS model has been fully incorporated into the Geant4 built-in physics module, g4em-standard_opt4, we switch to using this in our default physics list and remove the custom module.

**Improved: removed the G4RadioactiveDecay module from our Default Physics List** We have found that the G4RadioactiveDecay process sometimes causes errors such as:

```
G4Exception : de0001 issued by : G4AtomicTransitionManager::Shell()
No de-excitation for Z= 3  shellIndex= 2>= numberOfShells= 2 AtomicShell not_found
```

Since this module is not needed for most simulations, we have removed it from the default. If you really want this process, you can add it back to Ph/Default/Modules.

**Fixed: renamed surfaces of TsCylinder and TsSphere** We have revised the names of Surfaces to have a more consistent overall design. Phi/Theta now have Plus/Minus afterwards, like X/Y/Z.

- PlusPhiSurface -> PhiPlusSurface
- MinusPhiSurface -> PhiMinusSurface
- PlusThetaSurface -> ThetaPlusSurface
- MinusThetaSurface -> ThetaMinusSurface

**Fixed: corrected some surface area calculations**

- TsCylinder: calculation was wrong for area of Z surfaces and curved surfaces.
- TsSphere: calculation was wrong for area of curved surfaces when there was a phi cut.
- TsSphere: calculation was wrong for area of phi and theta cut surfaces.

We are sorry to have allowed these errors to slip through our testing process. Thanks to Christian Sommer for alerting us to the first of these (which led to a full review).

**Added: other minor features**

- We added a way to have TOPAS list all processes in the currently selected physics list:

```
:b:Ph/ListProcesses = "True"
```

- Topas can now tell you it’s version information: just type: topas --version
Fixed: other minor bugs

- Removed need for BeamEnergy and BeamEnergySpread when source is spectrum.
- Removed the particle source type Twiss as source type Emittance does the same and more.
- The angular generation for Beam sources has been corrected so that it is valid beyond the small-angle approximation.
- The base class for scorers, TsVScorer.hh, now includes G4SystemOfUnits.hh. Scoring often uses units, and this should make everyone’s life easier.
- We removed the requirement that some water be present in the simulation when using the DoseToWater and DoseToMaterial scorers.
- In time feature random time mode, Ts/ShowHistoryCountAtInterval now counts runs rather than events.
- The material name Flourine has been corrected to Fluorine.
- The parameter Ph/*/LamdaBins has been corrected to Ph/*/LambdaBins.
- PhaseSpaceBufferSize is now called OutputBufferSize. We renamed this parameter as it now applies not just to phase space but also to n-tuples.
- Sources now move correctly when the source component’s parent component is moving. This situation used to work only when the source component’s parent was a group component. It now works correctly for all cases. Thanks to Christian Sommer for showing us this bug.
- Phase space source now correctly handles all ions. Some ions were previously being forbidden in the phase space source. Thanks to Vadim Moskvin for reporting this bug.
- Solved bug that was causing part of phase space file to be used by two separate threads. When a multi-threaded session was using a phase space source, histories were being incorrectly assigned to the worker threads, causing some histories at the end of the file to be used more than once. Thanks to Hugo Moreira for showing us this bug.

2.0 Series

The main feature introduced by the 2.0 series was multithreading. In doing so we changed the underlying Geant4 to version Geant4.10.1.p02. This upgrade required extensive recoding within TOPAS, but was done in a way that required almost no changes to Parameter Files.

2.0.3 (2016-01-12)

Fixed: bug affecting scoring in divided TsBoxes, TsCylinders and TsSpheres This caused some of the dose due to secondary particles to be assigned to wrong divisions. This bug was introduced when we switched to the Geant4.10 series, so began at Topas 2.0.

Fixed: bug in DICOM output Dose value per voxel index was correct, but the entire structure was being drawn too small.

Fixed: bug using “Map” magnetic field type In multi-threaded mode, field only took effect after the first run.

Fixed: bug in Phase Space output Was crashing for cases of Ts/NumberOfThreads = 0 (meaning use all threads).

Fixed: a type in the header of DoseToWaterBinned scorer Header had some extraneous text left over from an earlier design.
**Improved:** restored two convenience methods to TsVScorer  
GetRunID and GetEventID are once again available for use by user-written scorers.

### 2.0.2 (2015-11-18)

**Fixed:** segfaults from secondary biasing  
Filtering on whether a particle interacted in a given component was failing to notice the interactions. Thanks to Gray Lu and Christian Sommer for reporting these bugs.

**Fixed:** first few histories could ignore filters when using multiple threads  
In testing our fix for the particle interaction filter, we also found a more subtle bug that could cause the first few histories to ignore any filter when running on more than one thread.

### 2.0.1 (2015-11-13)

**Fixed:** scorers report incorrect “Sum” in a specific case  
Bug triggered when you have more than one bin (that is, where XBin, YBin, ZBin, RBin, PhiBin or ThetaBin is more than one) and the scorer’s Report options are set to exactly “Sum” and “Mean”.

### 2.0.0 (2015-11-04)

**Added:** multithreading support  
Set $i$:Ts/NumberOfThreads to the number of CPU threads you want to use. If set to a positive integer, TOPAS will use that number of threads. If set to 0, TOPAS will use all of your computer’s threads (may be number of hardware cores or number of virtual cores (which includes hyper-threading cores) depending on your hardware architecture. If set to a negative number, TOPAS will use all BUT this number of threads, leaving you some threads reserved for other tasks (email, web browsing, etc.).

**Improved:** updated physics lists for Geant4.10.1.p02  
We have upgraded the default physics list to what we believe is the best option for proton therapy dose calculation in this Geant4 release. Remember that the default physics settings in TOPAS may not be the best settings for your own work. The only assurance we can give you is that the default settings are what we have currently chosen for proton therapy dose calculation research at our home institutions of MGH and UCSF. Geant4 physics changes in each release, and it is always the user’s responsibility to perform any validations and adjustments that may be required.

**Improved:** magnetic field is now just a parameter on any component  
You no longer specify special components for magnetic fields. Instead, the field is just an extra parameter that you can set for any of the standard components. So where you used to have:

```plaintext
Ge/MyComponent/Type = "TsDipoleMagnet" # or "TsQuadrupoleMagnet",
"TsTabulated3DField"
```

you now use:

```plaintext
Ge/MyComponent/Type = "TsBox" # or "TsCylinder", etc
Ge/MyComponent/MagneticField = "Dipole" # or "Quadrupole", "Map"
```

which gives more flexibility.

**Added:** quadrupole magnetic more flexible  
Can now have separate GradientX and GradientY.

**Added:** visualization of magnetic fields  
Field intensity and direction are represented as a set of arrows. The arrow density is controlled by:

```plaintext
i:Gr/ViewA/MagneticFieldArrowDensity = 10
```
Use with caution. When combined with rotation it sometimes causes crashes in polycone drawing (involved in drawing the arrowheads).

**Added:** magnetic fields to extensions mechanism  See here for how to add custom magnetic fields.

**Improved:** overlap checking is more strict  Geometry Overlaps previously caused only a warning. This meant that users might not even notice that their simulation had this dangerous geometry problem. TOPAS now is set to quit if any overlap is detected. If you really want TOPAS to continue, you can set:

```
Ge/QuitIfOverlapDetected = "False"
```

You will then still get a warning when the overlap is detected, an another warning at the end of the session. As before, you can turn off overlap checking entirely with:

```
Ge/CheckForOverlaps = "False"
```

This saves a small amount of time at startup, but is only recommended in cases where you are running a setup that you have already extensively tested.

**Added:** “Twiss” source type replaced by “Emittance”  The new source type provides more flexibility than the previous Twiss (see here).

**Improved:** user extensions require updating

- If you have written your own Scorer in C++, you will need to add one additional argument to the constructor and pass this argument on to the TsVScorer.
- If you have written your own Particle Source in C++, you will need to redesign this to have separate TsSource and TsGenerator.

**Fixed:** rare bug affecting some apertures  We have found rare cases in which an Aperture leaked dose (particles passed through one part of the aperture as material as if there was no material present). This was traced to an underlying bug in Geant4’s TessellatedSolid. While the bug is not yet fixed in Geant4, we now trap it and interrupt the relevant history. If this bug affects your session you will see warning message each time it occurs, plus a summary about this bug at the end of the console. An additional parameter aborts the session if this bug is found more than a specified number of times:

```
/i/Ts/MaxInterruptedHistories = 10 # defaults to 10
```

We do not recommend setting this to be a significant fraction of the total number of simulated histories.

**Added:** new examples  Random versus Sequential Time Feature modes are demonstrated in two new examples (*RunRandom_Mode.txt* and *RunSequential_Mode.txt*). Bremsstrahlung splitting is demonstrated in a new example (*SecondaryBiasing.txt*).

### 1.X Series

#### 1.3.0 (2015-10-21)

**Improved:** scoring report option “Min_Max” is split into “Min” and “Max”  If you have previously used the option “Min_Max”, you will need to replace it with “Min” “Max”.

**Added:** new ways to dump parameters to text files  *Ts/DumpParametersToSimpleFile* takes a list of parameter names and dumps their types, names and values to a simple, human-readable file.  *Ts/DumpParametersToSemicolonSeparatedFile* takes a list of parameter names and dumps their types, names and values to a semicolon separated file, suitable for easy import into other applications.
Added: method of defining elements  Elements can now be defined from specific combinations of isotopes (rather than from natural abundance).

Improved: zero padding of RunID in output file names  For example, MyScorerOutput_Run_0001.csv rather than MyScorerOutput_Run_1.csv, so that they will sort naturally in various file viewers. The number of padding places can be modified by the parameter i:Ts/ RunIDPadding.

Improved: scoring filters can now specify ion charges  Source ions must be fully stripped (as before), but this restriction is no longer applied to scoring filters. For example:

```plaintext
sv:Sc/IonsWithCharge3to5/OnlyIncludeParticlesNamed = 1 "GenericIon(6,12,5)"
```

will only score those carbon ions that have charge of 5.

Improved: console output of phase space scorers  Additional information reported to the console.

Improved: filtering example  Now includes InitialParticle filters (see examples/Scoring/ Filters.txt).

Improved: replaced the LET scorer  The new version has a new name, ProtonLET, to clarify that this calculation technique is only appropriate for protons. It uses a new “subscorer” mechanism by which the energy deposit and step length are internally scored by two separate scorers which are created for you on the fly and then divided to produce the final value.

Fixed: phi divisions in TsCylinder and TsSphere  The phi order of the divisions was previously backwards. For cases where the Cylinder or Sphere covered the full two Pi, this was only a numbering issue. But for cases where the Cylinder or Sphere was cut to less than the full two Pi, this caused the phi divisions to extend outside of the mother volume.

Fixed: excessive generation of parallel worlds  Some jobs were unnecessarily hitting the maximum number of parallel worlds (8). TOPAS was generating parallel worlds where they were not needed for one very specific situation: when a component’s name was a subset of a surface scorer’s target component’s name. So for example, when there was a component named MyComponent while a scorer was using MyComponent/ SubComponentName. (Yes, never mind the details, it’s fixed.)

Fixed: other minor bugs

- The phase space might not read all particles in the phase space file.
- Setting AssignToRegionNamed for a component that is unused creates a segfault.
- Using verbose scoring output from unsegmented components creates a segfault.
- Histogram IDs in xml and root files had incorrect run numbers.
- Quadrupole magnetic field incorrectly respected rotation of the component.

1.2.2 (2015-06-12)

Improved: TOPAS now prevents Geant4 from hanging due to stuck particles  We check the particle’s GetCurrentStepNumber at every step, killing the particle in the very rare case that the step numbers exceeds 100,000 (reasonable particles generally take no more than a few thousand steps). The limit can be adjusted by the parameter Ts/MaxStepNumber, but there is unlikely to be any reason that you should do so (stuck particles are rare, and even taking 100,000 steps doesn’t take very long). In principle, stuck particles should never occur, as Geant4 has a mechanism in place to gently nudge stuck particles, but this mechanism appears to have occasional failings (on the order a few per ten million histories). Because this mechanism now requires us to include a stepping action for every step in every region, we have checked carefully to evaluate its cost in CPU time. The cost appears to be only about 1%, well worth paying to avoid the workflow disruption that comes from stuck jobs. A message is printed at the bottom of the console if any particles have been killed by this new check. It tells the total number of particles affected and the total kinetic energy involved. More detailed information on each killed particle is provided earlier in the console log.
**Improved:** testing of complex parameter file graphs  These are complex arrangements with multiple `includeFile`. This fixes an issue in which our checks to prevent ambiguous parameter file graphs had been rejecting some valid graphs.

**Fixed:** correctly handle non-zero timeline starting value  This fixes an issue in which geometry setup failed or reported overlaps at time zero, when time zero was not actually going to be used for any of the runs.

**Fixed:** placement within parallel worlds  Components that are children of a parallel world group component are now placed correctly.

**Improved:** error messages from sources  Particle sources now have improved checks and more helpful error messages when required parameters are missing.

**Fixed:** bug in DVH output  DVH header files now give the correct name for their DVH data file. This fixes a minor issue in which the wrong file extension was shown in the header.

**Improved:** specifying ions  Ion handling has been updated to reflect that Geant4 only handles fully stripped ions. The third parameter in `GenericIon(Z, A, Charge)` is now optional. If present, it must equal Z since Geant4 only handles fully stripped ions. Since this argument is redundant, you can also just specify `GenericIon(Z, A)`.

**Added:** IncludeCharge option for phase space output  This may be informative for ions that can lose charge as they move through material. Not available for Limited output format.

### 1.2.1 (2015-05-15)

**Fixed:** bug in range modulator wheel component  The bug caused a mis-alignment in the position of the modulator wheel blocks. The magnitude of the effect depends on ones exact simulation (and hence was missed in our own SOBP tests). A review of changes shows that we introduced this bug just before we upgraded from our Beta releases to our Release 1.0. Thanks to Benjamin Lutz for identifying this bug.

**Fixed:** incorrect RunID output by parameter dump  Triggered by diagnostic parameter dumps from `Ts/DumpParametersToCsvFile`, etc.

**Fixed:** remove extraneous debugging messages  Removed diagnostic output that was left over from recent development of the beam source spectrum feature.

### 1.2.0 (2015-04-20)

**Added:** energy spectrum parameters to beam source  Beam particle source can sample energy from a user-defined spectrum [see here](#).

**Added:** particle sources to extension mechanism  See [here](#) for how to add custom particle sources.

**Added:** new user hooks to extension mechanism  See [here](#) for how to add new methods: `BeginSession`, `BeginRun`, `BeginHistory`, `EndHistory`, `EndRun`, `EndSession`.

**Improved:** column ordering in scorer output can be customized  Whereas previously, the Report parameter just indicated which quantities to report (Sum, Mean, etc.), the parameter is now also used to tell the order of these output columns. Thus: `sv:Sc/MyScorer/Report = 2 "Sum" "Mean"` now gives a different column order than `sv:Sc/MyScorer/Report = 2 "Mean" "Sum"`.

**Improved:** removed trailing comma in csv output format  The trailing comma on each line caused problems for some Matlab users.

**Added:** ability to read back and manipulate scorer output  Scoring Output can be read back in, so that one can then write out with different Report options:

```
Ts/RestoreResultsFromfile = "True"  # defaults to "False"
```
will then expect each scorer to have:

```plaintext
# :Sc/MyScorer1/InputFile = "MySavedFileName"
# :Sc/MyScorer1/InputType = "csv" # "csv" or "binary"
```

Reads header of input file to check that input is appropriate for the given scorer. Can be used to output with different Report options than the original file, such as different columns, different column order or creating a DVH. Can also be used to translate from csv to binary or back again.

**Added: option to dump parameter values to csv file** Dumps the requested parameters to a txt file: TopasParameter-Dump_Run0.txt. Triggered by:

```plaintext
#v :Ts/DumpParametersToCsv = 2 "SomeParameter" "SomeOtherParameter"
```

When multiple runs are involved, makes a new file for each run.

**Fixed: by default, phase spaces do not output empty histories** `PhaseSpaceIncludeEmptyHistories` default value changed to False to match what was already in the User Guide.

**Improved: meaning of EBins = 1 and TimeBins = 1 has changed** Previously, setting these values to 1 meant “do not use binning”. But since even when there is only one bin the data in the underflow and overflow bins can be useful, we now have EBins = 1 mean create one bin, plus underflow and overflow bins. To have no binning at all, set EBins = 0 (or don’t set EBins at all). A similar change was made for TimeBins = 1.

**Fixed: conflict between energy binning and some filters** Energy Binning was not working in the presence of `FilterByAtomicMass`, `FilterByAtomicNumber` and the example user-supplied filter, `TsMyFilter1`.

**Fixed: memory bug in energy binning** Previously caused segfault on some systems when EBins was used.

**Improved: check vector parameter length is non-negative** Previously protected against this number being zero but not against this being negative.

**Added: example of rotation directions** See `examples/TimeFeature/Rotation.txt`.

**Fixed: size of parallel scoring component for DICOM and XiO patient** This is the parallel world `TsBox` that is created if you attempt to score a DICOM or XiO Patient in a different grid than the original input file.

**Fixed: visualization of parallel scoring component for DICOM and XiO patient** These parallel scoring components were often invisible. They now show correctly.

### 1.1.0 (2015-02-23)

**Fixed: bug in particle type filters** It corrects behavior of the following four types of filters:

- `OnlyIncludeParticlesNamed`
- `OnlyIncludeParticlesNotNamed`
- `OnlyIncludeIfParticleOrAncestorNamed`
- `OnlyIncludeIfParticleOrAncestorNotNamed`

### 1.0.0 (2015-02-11)

First version of TOPAS was released! (Uses Geant4.9.6.p04)

Prior to the release of version 1.0, TOPAS had a long public beta phase.
SECTION 7

Introduction to Parameter System

Design Philosophy

TOPAS follows a consistent set of design paradigms. Understanding these paradigms will make TOPAS more intuitive to you.

All control is through the TOPAS Parameters System. Use of Geant4 macros or interactive commands is not supported as it does not give you the reliability and repeatability that comes from the parameters system.

• TOPAS Parameter files are not Geant4 macro files. TOPAS is specifically designed to avoid the kind of order-dependence risks that Geant4 macro files create.

• TOPAS Parameter files are not XML files. Those too involve the kind of order dependence that we explicitly avoid.

To keep OpenGL graphics from vanishing from the screen, you have the option to have TOPAS pause at the Geant4 command line by including the option:

\[b:Ts/PauseBeforeQuit = \text{"True"} \quad \# \text{ defaults to "False"}\]

To exit the Geant4 command line, and continue with the TOPAS session, type `exit`. For simulations that do not involve OpenGL graphics, just leave this option at "False".

All positions are set relative to Geometry Components. If you want to place a particle source or a scorer, you place it relative to a particular Component. You may choose to do your placement relative to the center of the World component, in which case you have essentially used the global coordinate system, but you will likely choose a more directly relevant Component. For example, a source that represents a particle beam might be placed at the beamline exit window. Doing so means that the source position will move appropriately with any nozzle movement.

All time dependent behaviors are controlled through the Time Features system.

TOPAS fully supports the Multi-Threaded simulation capability of Geant4. By default, TOPAS will occupy just one CPU thread, but you can spread the simulation over multiple threads, distributing the load over your entire computer, by adjusting the parameter `i:Ts/NumberOfThreads`. See details in the section Multithreading.
Syntax

The TOPAS Parameter System is a control structure for applications in which a large number of complex inter-related parameters are controllable by designers and end-users, in a manner that is absolutely flexible but simultaneously easy to use. The system is designed with safety and repeatability as top priorities. A key error-checking strategy is strict type checking, in which every parameter must have a specific declared type (string, boolean, integer, etc.) and the provided values are checked to ensure they are appropriate to the given type.

The system takes a set of “Parameters Files,” simple text files made up of lines of key/value pairs:

```
Parameter_Type : Parameter_Name = Parameter_Value # Optional comment
```

When you edit parameter files, be careful to use a Plain Text editor. TOPAS will not understand the various hidden characters created by complex word processors (such as Word or Keynote). Whatever your editor, turn off advanced features such as “Smart quotes”, “Smart dashes” and “Smart links”.

Ten example parameter settings are given below:

```
d:Ge/Phantom/HLX = 10. cm # Dimensioned Double
u:Ge/Magnet/Dipole/MagneticFieldDirectionX = 1.0 # Unitless Double
i:Sc/DoseScorer/ZBins = 100 # Integer
b:Sc/DoseScorer/Active = "True" # Boolean
s:Ge/Phantom/Material = "G4_WATER" # String
dv:Ge/RMW_Track1/Angles = 4 69.1 92.2 111.0 126.0 deg # Dimensioned
  → Double Vector
uv:Ma/Phantom_Plastic/Fractions = 3 0.05549 0.75575 0.18875 # Unitless Double
  → Vector
iv:Gr/Color/yellow = 3 225 255 0 # Integer Vector
bv:Tf/ScoringOnOff/Values = 4 "true" "false" "true" "false" # Boolean Vector
sv:Ma/MyPlastic/Components = 3 "Hydrogen" "Carbon" "Oxygen" # String Vector
```

Note: The order of lines within a parameter file does not matter.

A Parameter_Name can be almost any string, but we have prefix conventions to keep things clear:

- **Ma**/ for Materials
- **El**/ for Elements
- **Is**/ for Isotopes
- **Ge**/ for Geometry Components
- **So**/ for Particle Sources
- **Ph**/ for Physics
- **Vr**/ for Variance Reduction
- **Sc**/ for Scoring
- **Gr**/ for Graphics
- **Tf**/ for Time Features
- **Ts**/ for TOPAS overall control

The Parameter_Type tells TOPAS what type of data will be in this parameter:

- **d** for Dimensioned Double
TOPAS uses this Parameter_Type to perform “strict type checking,” checking that the Parameter_Value is appropriate and complete for the given Parameter_Type.

A String parameter must be in quotes and may take any value.

A Boolean parameter must be in quotes and may be either:
- "True", "t" or "1" (in any case) to mean true
- "False", "f" or "0" (in any case) to mean false

An Integer parameter must be something that can be interpreted as an integer.
- The value may not contain any decimal part, as this can lead to ambiguity as to the employed rounding strategy.
- These are 32 bit integers, thus the values can range from 0 to 2147483647.

A Dimensioned Double parameter requires both a value and a unit.
- We require the unit to avoid misunderstandings.
- The value must be something that can be interpreted as a floating point number.

A Vector of Dimensioned Doubles parameter requires an integer (larger than zero) to indicate how many values are expected, then the values themselves, then a unit.
- Vector of Dimensioned Doubles is useful when the definition of a single shape, motion, etc. requires multiple dimensioned double values.
- Our usage of the term “vector” may be unfamiliar to some readers but is the standard term for such structures in modern programming languages.

Vectors of Unitless, Integer, Boolean and String again require an integer to indicate how many values are expected, then the values themselves. The individual strings in a Vector of Strings can not contain spaces (this requirement will be relaxed in a subsequent TOPAS release).

The comment character is #. Anything to the right of the comment character is taken as a comment. Comments can span as many lines as desired, until a new line is found that contains the equals sign.

Blank lines are ignored.

Parameter names may use mixed case, but their interpretation is not case sensitive. That is, "myParameter" is considered the same as "myParameter" or "myPaRaMeter", etc.
Complete Set of Allowed Syntax for any one Parameter Line

Warning: Note that in all of the expressions below, there must be a space before and after any +, − or *.

Dimensioned Double parameters:

```
d:parameterName = number unit
```

```
d:parameterName = number unit + name_of_dimensioned_double_parameter
```

```
d:parameterName = number unit − name_of_dimensioned_double_parameter
```

```
d:parameterName = number unit * name_of_unitless_or_integer_parameter
```

```
d:parameterName = name_of_dimensioned_double_parameter unit
```

```
d:parameterName = name_of_dimensioned_double_parameter unit * number
```

```
d:parameterName = name_of_unitless_or_integer_parameter * number unit
```

```
d:parameterName = name_of_unitless_or_integer_parameter + number unit
```

```
d:parameterName = name_of_unitless_or_integer_parameter - number unit
```

```
d:parameterName = name_of_unitless_or_integer_parameter * number unit
```

```
d:parameterName = name_of_unitless_or_integer_parameter + name_of_unitless_or_integer_parameter
```

```
d:parameterName = name_of_unitless_or_integer_parameter - name_of_unitless_or_integer_parameter
```

```
d:parameterName = name_of_unitless_or_integer_parameter * name_of_unitless_or_integer_parameter
```

Unitless parameters:

```
u:parameterName = number
```

```
u:parameterName = number + name_of_unitless_or_integer_parameter
```

```
u:parameterName = number − name_of_unitless_or_integer_parameter
```

```
u:parameterName = number * name_of_unitless_or_integer_parameter
```

```
u:parameterName = name_of_unitless_or_integer_parameter
```

```
u:parameterName = name_of_unitless_or_integer_parameter + integer
```

```
u:parameterName = name_of_unitless_or_integer_parameter − integer
```

```
u:parameterName = name_of_unitless_or_integer_parameter * integer
```

```
u:parameterName = name_of_unitless_or_integer_parameter + name_of_unitless_or_integer_parameter
```

```
u:parameterName = name_of_unitless_or_integer_parameter − name_of_unitless_or_integer_parameter
```

```
u:parameterName = name_of_unitless_or_integer_parameter * name_of_unitless_or_integer_parameter
```

Integer parameters:

```
i:parameterName = integer
```

```
i:parameterName = integer + name_of_integer_parameter
```

```
i:parameterName = integer − name_of_integer_parameter
```

```
i:parameterName = integer * name_of_integer_parameter
```

```
i:parameterName = name_of_integer_parameter
```

```
i:parameterName = name_of_integer_parameter + integer
```

```
i:parameterName = name_of_integer_parameter − integer
```

```
i:parameterName = name_of_integer_parameter * integer
```

```
i:parameterName = name_of_integer_parameter + name_of_integer_parameter
```

```
i:parameterName = name_of_integer_parameter − name_of_integer_parameter
```

```
i:parameterName = name_of_integer_parameter * name_of_integer_parameter
```

Boolean parameters:
String parameters:

```
:s:parameterName = string
:s:parameterName = string + name_of_integer_or_string_parameter
:s:parameterName = name_of_integer_or_string_parameter
:s:parameterName = name_of_integer_or_string_parameter + string
:s:parameterName = name_of_integer_or_string_parameter + name_of_integer_or_string_parameter
```

Dimensioned Double Vector parameters:

```
dv:parameterName = number_of_values value1 value2 ... valueN unit
dv:parameterName = number_of_values value1 value2 ... valueN unit + name_of_dimensioned_double_or_double_vector_parameter
dv:parameterName = number_of_values value1 value2 ... valueN unit - name_of_dimensioned_double_or_double_vector_parameter
dv:parameterName = number_of_values value1 value2 ... valueN unit * name_of_unitless_or_integer_or_unitless_vector_or_integer_vector
```

Unitless Vector parameters:

```
:uv:parameterName = number_of_values value1 value2 ... valueN
:uv:parameterName = number_of_values value1 value2 ... valueN + name_of_unitless_or_integer_or_unitless_vector_or_integer_vector
:uv:parameterName = number_of_values value1 value2 ... valueN - name_of_unitless_or_integer_or_unitless_vector_or_integer_vector
:uv:parameterName = number_of_values value1 value2 ... valueN * name_of_unitless_or_integer_or_unitless_vector_or_integer_vector
:uv:parameterName = name_of_unitless_vector_parameter
:uv:parameterName = integer * name_of_unitless_vector_parameter
:uv:parameterName = name_of_unitless_or_integer_parameter * name_of_unitless_vector_parameter
```

Integer Vector parameters:

```
i:parameterName = number_of_values value1 value2 ... valueN
i:parameterName = number_of_values value1 value2 ... valueN + name_of_integer_or_integer_vector_parameter
i:parameterName = number_of_values value1 value2 ... valueN - name_of_integer_or_integer_vector_parameter
i:parameterName = number_of_values value1 value2 ... valueN * name_of_integer_or_integer_vector_parameter
i:parameterName = name_of_integer_vector_parameter
i:parameterName = integer * name_of_integer_vector_parameter
```

# value1, value2, etc. can be a numeric value or the name of a dimensioned double or double vector parameter.
Boolean Vector parameters:

```plaintext
bv:parameterName = number_of_values value1 value2 ... valueN
bv:parameterName = name_of_boolean_vector_parameter
# value1, value2, etc. can be a numeric value or the name of a boolean parameter
```

String Vector parameters:

```plaintext
sv:parameterName = number_of_values value1 value2 ... valueN
sv:parameterName = number_of_values value1 value2 ... valueN + name_of_integer_or_
˓→string_or_integer_vector_or_string_vector
sv:parameterName = name_of_string_vector_parameter
# value1, value2, etc. can be a numeric value or the name of a string parameter
```

Other operations are intentionally not supported since their behavior might be unclear. Such things can be done in user C++ code, generating new parameters on the fly (see Transient Parameters). \( d \times d \) is forbidden because can create new units that we don’t recognize. Division is forbidden because of divide by zero issues, etc.

### Relative Parameters

TOPAS supports “relative parameters”, wherein one parameter may be set relative to another, as in:

```plaintext
s:Ge/Phantom/Material = SomeOtherParameterName
```

**Note:** The many uses of this relative parameter syntax become more clear once one understands the entirety of the TOPAS design, including hierarchical control files and time features.

With relative dimensioned double parameters, we must protect against a user setting a parameter relative to some other parameter that does not have appropriate units. The solution is to insist that a unit be included on the right side of the expression. In the example below, the unit of \( \text{cm} \) indicates that SomeOtherParameter must itself have units of length.

If that other parameter’s unit is of the entirely wrong unit category (mass, angle, etc.), TOPAS will refuse to run. If the unit is of the right category but a different exact unit (m, mm, etc.), TOPAS will perform appropriate unit conversion:

```plaintext
d:Ge/Phantom/HLX = SomeOtherParameterName cm
```

TOPAS has a grammar for operations such as adding or multiplying parameters:

```plaintext
Ge/Compensator/TransZ = Ge/Aperture/DistalEdge + Ge/Compensator/HLZ mm
```

**Warning:** Note that there must be a space before and after the plus sign.

Relative parameters allow only a limited number of functions, intentionally not a full math library, since other math functions may be ambiguous, requiring too much prior understanding of the mathematical syntax. The complete set of allowed syntax for any one parameter line is shown here.
Hierarchical Control

Parameter files may pull in other parameters through `includeFile` statements, such as:

```plaintext
includeFile = someOtherParameterFile
```

On most operating systems, the case of the file name matters - `MyIncludeFile.txt` is not considered the same file as `MYIncludeFile.txt` - so take care to match the exact case.

`includeFile` lines may appear anywhere in the parameter file.

`includeFile` can use either absolute or relative file positions, such as:

```plaintext
includeFile = /Applications/topas/someDirectory/MyIncludeFile.txt
# or
includeFile = ../someDirectory/MyIncludeFile.txt
```

A file inherits all settings from its `includeFile` statements, and can override any of those included settings by setting the same parameter name to a new value.

Type and kind of units has to match type and kind of units for the same parameter name in any `includeFile`. Type can be omitted if the same parameter name has already been defined with a type in an `includeFile`.

**Parameter File Chains**

When a parameter file includes another parameter file, and this in turn includes another parameter file, we refer to this as a “parameter file chain”.

Parameter file chains fit nicely into research workflow. You can define most of your standard settings in one file, while a file higher on the chain overrides just those values that you want to change today.
To define a parameter in terms of the value of the same parameter in an includeFile, set the value to the same parameter name (always interpreted to mean this parameter value from an includeFile) or use the shortcut value, inheritedValue. The following example would set a foil to be twenty percent thicker than in its included file:

```
Ge/IonChamber/Layer2/Foil/HLZ = Ge/IonChamber/Layer2/Foil/HLZ mm * 1.2
# or
Ge/IonChamber/Layer2/Foil/HLZ = inheritedValue mm * 1.2
```

**Warning:** Note that there must be a space before and after the multiplication sign.

A basic set of TOPAS default parameters are built into the system (see Default Parameters). You may override these defaults in your own parameter files if you wish.

**Parameter File Graphs**

TOPAS is designed to facilitate multiple independent workgroups focused on separate aspects such as treatment head design, patient handling and imaging devices. To this end, a parameter file may inherit settings from more than one other parameter file, forming a structure that is more complex than just a single chain. We refer to such a structure as a “parameter file graph.” An example is shown below:

To implement such designs, a parameter file allows any number of includeFile statements, the statements may be located anywhere in the file, and you may specify one or more include files in a single includeFile statement, as in:

```
includeFile = someFile someOtherFile someOtherOtherFile
includeFile = stillAnotherFile
```
There is no significance to the left to right arrangement. That is:

```
includeFile = fileA fileB fileC
```

will behave the same as:

```
includeFile = fileC fileB fileA
```

or even the same as if this was broken up into multiple `includeFile` statements:

```
includeFile = fileB
includeFile = fileC fileA
```

To keep this order from mattering, TOPAS does not allow you to set up a simulation object (a Geometry Component, a Scorer, etc.) in one chain but modify it in a different chain. All parameters on a given Geometry Component need to be handled in the same chain. All parameters on a given Scorer need to be handled in the same chain.

TOPAS also checks to make sure that no two parameter file chains modify the same parameter in a way that is ambiguous. If, for example, the material MySpecialTungstenAlloy has been defined in the imaging chain, it cannot also be defined differently in the treatment head chain (unless the top level file, the user file, itself defines this parameter in an absolute way, that is, not relative to any other parameter).

The basic paradigm controlling use of multiple parameter chains is that nothing you do in one chain should magically change the behavior of anything in another chain. Think of the simulation world as an actual physical room. We don’t want the behavior of one thing in the room to magically change just because something else entered the room. So when you have multiple chains, TOPAS will check that neither chain modifies anything from the other chain.

This also means that no chain can redefine any of the Default Parameters, since all of the Default Parameters effectively belong at the base of every chain. The only place you can redefine the Default Parameters is in the top parameter file. This can feel like a heavy requirement, but it is essential. We want TOPAS to be a great tool for use in collaborative research environments where several people or teams may be contributing their own parts of the simulation setup.

It can take some experience to design complex parameter file chains. If you get stuck, feel free to ask for help on the TOPAS User Forum.

## Controlling Multiple Batch Jobs

The hierarchical nature of parameter files makes it easy to control multiple batch jobs.

Make up a parameter file (or hierarchy of files) that has most of your settings:

- MostOfMySettings.txt

Then make small additional parameter files for each job you want to submit:

- Job1.txt
- Job2.txt
- Job3.txt

where each of these files has:

```
includeFile = MostOfMySettings.txt
Ts/Seed = 1 # Set this differently for each of Job1, Job2, Job3, ...
Sc/MyScorer/OutputFile = "Job1Output" # Set this differently for each of Job1, Job2, ...
```

7.5. Controlling Multiple Batch Jobs
Each job will thus have a unique starting random number seed (and hence produce a statistically distinct sample) and a unique output file specification, but all other aspects of the simulation will be identical from one job to the next.
The following parameters are built-in by default. They are actually compiled into the code rather than set from a parameter file, so that all users will always have the same starting set of defaults. You can override any of these parameters in your own files.

**Overall program control**

```plaintext
i:Ts/NumberOfThreads = 1 # Number of CPU threads that work will be distributed to
b:Ts/BufferThreadOutput = "False" # Causes console output to be show one thread at a
˓→time
i:Ts/Seed = 1 # starting random seed
i:Ts/MaxStepNumber = 100000 # limit on number of steps before a track is killed
i:Ts/MaxInterruptedHistories = 10 # limit on how many histories can throw rare Geant4
˓→errors
b:Ts/DumpParameters = "False" # Set true to dump full set of parameters to html file
˓→TopasParameterDump_Run0.html
b:Ts/DumpNonDefaultParameters = "False" # Like above but omits defaults
b:Ts/ListUnusedParameters = "False" # Set true to list unused parameters on the
˓→console
i:Ts/ShowHistoryCountAtInterval = 1 # How often to print history count to the console
b:Ts/ShowHistoryCountOnSingleLine = "False" # Set true to make history count reuse
˓→same line of console
i:Ts/RunIDPadding = 4 # pad Run ID numbers to this many places in file names
b:Ts/PauseBeforeInit = "False" # Pause for Geant4 commands before initialization
b:Ts/PauseBeforeSequence = "False" # Pause for Geant4 commands before run sequence
b:Ts/PauseBeforeQuit = "False" # Pause for Geant4 commands before quitting
i:Ts/EventVerbosity = 0 # Set to larger integer to see details of run. Maximum is 2
i:Ts/TrackingVerbosity = 0 # Set to larger integer to see details of event. Maximum is 5
i:Ts/SequenceVerbosity = 0 # Set to larger integer to see details of TOPAS run
˓→sequence
```
Overall timeline control

```
b:Tf/RandomizeTimeDistribution = "False"
d:Tf/TimelineStart = 0. s
d:Tf/TimelineEnd = Tf/TimelineStart s
i:Tf/NumberOfSequentialTimes = 1
i:Tf/Verbosity = 0 # set to 1 to generate time log, set to 2 to get detailed update
   messages
```

Optional checks on correctness of geometry

```
b:Ge/CheckForOverlaps = "True"
b:Ge/QuitIfOverlapDetected = "True"
b:Ge/CheckForUnusedComponents = "True"
```

Top level geometry component, the World Volume

```
s:Ge/World/Type = "TsBox"
s:Ge/World/Material = "Air"
d:Ge/World/HLX = 5. m # Half Length
d:Ge/World/HLY = 5. m
d:Ge/World/HLZ = 5. m
d:Ge/World/TransX = 0. m
d:Ge/World/TransY = 0. m
d:Ge/World/TransZ = 0. m
d:Ge/World/RotX = 0. deg
d:Ge/World/RotY = 0. deg
d:Ge/World/RotZ = 0. deg
```

Demo Beam position

```
s:Ge/BeamPosition/Parent = "World"
s:Ge/BeamPosition/Type = "Group"
d:Ge/BeamPosition/TransX = 0. m
d:Ge/BeamPosition/TransY = 0. m
d:Ge/BeamPosition/TransZ = Ge/World/HLZ m
d:Ge/BeamPosition/RotX = 180. deg
d:Ge/BeamPosition/RotY = 0. deg
d:Ge/BeamPosition/RotZ = 0. deg
```

Demo Particle Source

```
s:So/Demo/Type = "Beam" # Beam, Isotropic, Emittance or PhaseSpace
s:So/Demo/Component = "BeamPosition"
s:So/Demo/BeamParticle = "proton"
s:So/Demo/BeamEnergy = 169.23 MeV
```
Physics

```
;So/Demo/BeamEnergySpread = 0.757504
;So/Demo/BeamPositionDistribution = "Gaussian" # Flat or Gaussian
;So/Demo/BeamPositionCutoffShape = "Ellipse" # Point, Ellipse, Rectangle or Isotropic
;So/Demo/BeamPositionCutoffX = 10. cm
;So/Demo/BeamPositionCutoffY = 10. cm
;So/Demo/BeamPositionSpreadX = 0.65 cm
;So/Demo/BeamPositionSpreadY = 0.65 cm
;So/Demo/BeamAngularDistribution = "Gaussian" # Flat or Gaussian
;So/Demo/BeamAngularCutoffX = 90. deg
;So/Demo/BeamAngularCutoffY = 90. deg
;So/Demo/BeamAngularSpreadX = 0.0032 rad
;So/Demo/BeamAngularSpreadY = 0.0032 rad
;So/Demo/NumberOfHistoriesInRun = 0
```

Scoring

```
;Sc/RootFileName = "topas" # name for root output files
;Sc/XmlFileName = "topas" # name for xml output files
```

Graphics

```
;Gr/Enable = "True" # Set False to avoid instantiating any part of Geant4
# visualization system (useful for running on batch machines that lack the OpenGL
# graphics library)
;Gr/Verbosity = 0 # Set to higher integer to increase verbosity of Geant4
# visualization system
;Gr/RefreshEvery = "Run" # "History", "Run" or "Session"
;Gr/ShowOnlyOutlineIfVoxelCountExceeds = 8000 # Above this limit, only show outer box
;Gr/SwitchOGLtoOGLIifVoxelCountExceeds = 3000000 # Above this limit, switch OpenGL
# Graphics to Immediate mode
```
Elements

Element Symbols:
- Hydrogen: H
- Helium: He
- Lithium: Li
- Beryllium: Be
- Boron: B
- Carbon: C
- Nitrogen: N
- Oxygen: O
- Fluorine: F
- Neon: Ne
- Sodium: Na
- Magnesium: Mg
- Aluminum: Al
- Silicon: Si
- Phosphorus: P
- Sulfur: S
- Chlorine: Cl
- Argon: Ar
- Potassium: K
- Calcium: Ca
- Scandium: Sc
- Titanium: Ti
- Vanadium: V
- Chromium: Cr
- Manganese: Mn
- Iron: Fe
- Cobalt: Co
- Nickel: Ni
- Copper: Cu
- Zinc: Zn
- Gallium: Ga
- Germanium: Ge
- Arsenic: As
- Selenium: Se
- Bromine: Br
- Krypton: Kr
- Rubidium: Rb
- Strontium: Sr
- Yttrium: Y
- Zirconium: Zr
- Niobium: Nb
- Molybdenum: Mo
- Technetium: Tc
- Ruthenium: Ru
- Rhodium: Rh
- Palladium: Pd
- Silver: Ag
- Cadmium: Cd
- Indium: In
- Tin: Sn
- Antimony: Sb
- Tellurium: Te
- Iodine: I
- Xenon: Xe
- Caesium: Cs
- Barium: Ba
## Materials

```plaintext
Ma/DefaultColor = "white"
Ma/Verbosity = 0 # Set to 1 to report each time a material is defined

sv:Ma/Vacuum/Components = 4 "Carbon" "Nitrogen" "Oxygen" "Argon"
uv:Ma/Vacuum/Fractions = 4 0.000124 0.755268 0.231781 0.012827
d:Ma/Vacuum/Density = 1.0E-25 g/cm3
sv:Ma/Vacuum/State = "Gas"
d:Ma/Vacuum/Temperature = 2.73 kelvin
d:Ma/Vacuum/Pressure = 3.0E-18 pascal
sv:Ma/Vacuum/DefaultColor = "skyblue"

sv:Ma/Carbon/Components = 1 "Carbon"
uv:Ma/Carbon/Fractions = 1 1.0
d:Ma/Carbon/Density = 1.867 g/cm3
d:Ma/Carbon/MeanExcitationEnergy = 78 eV
sv:Ma/Carbon/DefaultColor = "green"

sv:Ma/Aluminum/Components = 1 "Aluminum"
uv:Ma/Aluminum/Fractions = 1 1.0
d:Ma/Aluminum/Density = 2.6989 g/cm3
```
```
$s:Ma/Aluminum/DefaultColor = "skyblue"
$i:Ma/Aluminum/AtomicNumber = 13
$d:Ma/Aluminum/AtomicMass = 26.98154 g/mole

$s:Ma/Nickel/Components = 1 "Nickel"
$uv:Ma/Nickel/Fractions = 1 1.0
$d:Ma/Nickel/Density = 8.902 g/cm3
$s:Ma/Nickel/DefaultColor = "indigo"

$s:Ma/Copper/Components = 1 "Copper"
$uv:Ma/Copper/Fractions = 1 1.0
$d:Ma/Copper/Density = 8.96 g/cm3
$s:Ma/Copper/DefaultColor = "orange"

$s:Ma/Iron/Components = 1 "Iron"
$uv:Ma/Iron/Fractions = 1 1.0
$d:Ma/Iron/Density = 7.87 g/cm3
$s:Ma/Iron/DefaultColor = "skyblue"

$s:Ma/Tantalum/Components = 1 "Tantalum"
$uv:Ma/Tantalum/Fractions = 1 1.0
$d:Ma/Tantalum/Density = 16.654 g/cm3
$s:Ma/Tantalum/DefaultColor = "indigo"

$s:Ma/Lead/Components = 1 "Lead"
$uv:Ma/Lead/Fractions = 1 1.0
$d:Ma/Lead/Density = 11.35 g/cm3
$i:Ma/Lead/AtomicNumber = 82
$d:Ma/Lead/AtomicMass = 207.19 g/mole
$d:Ma/Lead/MeanExcitationEnergy = 823 eV
$s:Ma/Lead/DefaultColor = "brown"

$s:Ma/Air/Components = 4 "Carbon" "Nitrogen" "Oxygen" "Argon"
$uv:Ma/Air/Fractions = 4 0.000124 0.755268 0.231781 0.012827
$d:Ma/Air/Density = 1.20484 mg/cm3
$d:Ma/Air/MeanExcitationEnergy = 85.7 eV
$s:Ma/Air/DefaultColor = "lightblue"

$s:Ma/Brass/Components = 2 "Copper" "Zinc"
$uv:Ma/Brass/Fractions = 2 0.7 0.3
$d:Ma/Brass/Density = 8.550 g/cm3
$d:Ma/Brass/MeanExcitationEnergy = 324.4 eV
$s:Ma/Brass/DefaultColor = "grass"

$s:Ma/Lexan/Components = 3 "Hydrogen" "Carbon" "Oxygen"
$uv:Ma/Lexan/Fractions = 3 0.055491 0.755751 0.188758
$d:Ma/Lexan/Density = 1.2 g/cm3
$d:Ma/Lexan/MeanExcitationEnergy = 73.1 eV
$s:Ma/Lexan/DefaultColor = "grey"

$s:Ma/Lucite/Components = 3 "Hydrogen" "Carbon" "Oxygen"
$uv:Ma/Lucite/Fractions = 3 0.080538 0.599848 0.319614
$d:Ma/Lucite/Density = 1.190 g/cm3
$d:Ma/Lucite/MeanExcitationEnergy = 74.0 eV
$s:Ma/Lucite/DefaultColor = "grey"

$s:Ma/Mylar/Components = 3 "Hydrogen" "Carbon" "Oxygen"
$uv:Ma/Mylar/Fractions = 3 0.041959 0.625017 0.333025
```
**TOPAS Documentation, Release 3.0**

```
d:M/Mylar/Density = 1.40 g/cm³
s:M/Mylar/DefaultColor = "red"
sv:M/Mylon/Components = 4 "Hydrogen" "Carbon" "Nitrogen" "Oxygen"
v:M/Mylon/Fractions = 4 0.097976 0.636856 0.123779 0.141389
d:M/Mylon/Density = 1.140 g/cm³
d:M/Mylon/DefaultColor = "purple"
sv:M/Kapton/Components = 4 "Hydrogen" "Carbon" "Nitrogen" "Oxygen"
v:M/Kapton/Fractions = 4 0.026362 0.691133 0.073270 0.209235
d:M/Kapton/Density = 1.420 g/cm³
d:M/Kapton/DefaultColor = "purple"
sv:M/Water_75/Components = 2 "Hydrogen" "Oxygen"
v:M/Water_75/Fractions = 2 0.111894 0.888106
d:M/Water_75/Density = 1.0 g/cm³
d:M/Water_75/MeanExcitationEnergy = 75.0 eV
d:M/Water_75/DefaultColor = "blue"
sv:M/Titanium/Components = 1 "Titanium"
v:M/Titanium/Fractions = 1 1.0
d:M/Titanium/Density = 4.54 g/cm³
d:M/Titanium/DefaultColor = "blue"
sv:M/Steel/Components = 8 "Carbon" "Silicon" "Phosphorus" "Sulfur" "Chromium" "Manganese" "Iron" "Nickel"
v:M/Steel/Fractions = 8 0.0015 0.01 0.00045 0.0003 0.19 0.02 0.67775 0.1
d:M/Steel/Density = 8.027 g/cm³
d:M/Steel/DefaultColor = "lightblue"
```

**Colors**

```
iv:Gr/Color/lightblue = 3 175 255 255
iv:Gr/Color/skyblue = 3 175 124 255
iv:Gr/Color/red = 3 255 0 12
iv:Gr/Color/magenta = 3 255 0 255
iv:Gr/Color/violet = 3 224 0 255
iv:Gr/Color/pink = 3 255 0 255
iv:Gr/Color/indigo = 3 0 0 180
iv:Gr/Color/grass = 3 0 239 0
iv:Gr/Color/orange = 3 241 224 0
iv:Gr/Color/purple = 3 187 28 0
iv:Gr/Color/brown = 3 225 126 66
iv:Gr/Color/grey = 3 210 213 210
iv:Gr/Color/blue = 3 0 0 255
iv:Gr/Color/green = 3 0 126 126
iv:Gr/Color/yellow = 3 225 255 0
iv:Gr/Color/white = 3 255 255 255
iv:Gr/Color/black = 3 0 0 0
iv:Gr/Color/grey020 = 3 20 20 20
iv:Gr/Color/grey040 = 3 40 40 40
iv:Gr/Color/grey060 = 3 60 60 60
iv:Gr/Color/grey080 = 3 80 80 80
iv:Gr/Color/grey100 = 3 100 100 100
iv:Gr/Color/grey120 = 3 120 120 120
```

8.12. Colors
<table>
<thead>
<tr>
<th>Key</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>iv:Gr/Color/grey140</code></td>
<td>3 140</td>
<td>140</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td><code>iv:Gr/Color/grey160</code></td>
<td>3 160</td>
<td>160</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td><code>iv:Gr/Color/grey180</code></td>
<td>3 180</td>
<td>180</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td><code>iv:Gr/Color/grey200</code></td>
<td>3 200</td>
<td>200</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td><code>iv:Gr/Color/grey220</code></td>
<td>3 220</td>
<td>220</td>
<td>220</td>
<td></td>
</tr>
<tr>
<td><code>iv:Gr/Color/grey240</code></td>
<td>3 240</td>
<td>240</td>
<td>240</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 9

Overall Control

Time mode

If you do nothing special, TOPAS will do a single run with no time variation. We call this “Fixed Time Mode”. Other available modes are “Sequential” and “Random”.

Fixed Time Mode

To run in Fixed Time Mode, just set your source’s NumberOfHistoriesInRun, as in:

```plaintext
i:So/MySource/NumberOfHistoriesInRun = 100
```

If your parameter files include Time Features, they will all be evaluated with time equals zero. To instead have them evaluated at a different fixed time, specify TimelineStart, as in:

```plaintext
d:Tf/TimelineStart = 10. s # defaults to zero
```

If you have more than one source, the run will continue until all sources have run all of their histories. For each Geant4 “beamOn”, each source will get called, but only those that have more histories left to produce will actually produce any.

Sequential Time Mode

To have TOPAS do several runs at fixed time intervals, specify the start time, end time and number of sequential times, as in:

```plaintext
d:Tf/TimelineStart = 0. s # defaults to zero
d:Tf/TimelineEnd = 10. s # must be larger than TimelineStart
i:Tf/NumberOfSequentialTimes = 100 # defaults to 1
```

TOPAS will divide the overall time, TimelineEnd – TimelineStart, by NumberOfSequentialTimes and perform runs at each of these intervals.
The first run will be at time = TimelineStart.
The last run will be at time = TimelineEnd minus one interval. That is, TOPAS will stop before it reaches TimelineEnd.

So, in the example above:
- Run 0 will have Time = 0. s
- Run 1 will have Time = 0.1 s
- ...
- Run 99 will have Time = 9.9 s

At each of these intervals, your source will generate your indicated NumberOfHistoriesInRun:

| Source/MySource/NumberOfHistoriesInRun = 10 |

So, for example, if you have 100 intervals, and NumberOfHistoriesInRun = 10, you will generate a total of 100 x 10 = 1000 histories.

To have TOPAS print time feature information to a log file and to the console:

| Tf/Verbosity = 2  # defaults to zero. |
| # set to 1 to get time log (NbParticlesInTime.txt) |
| # set to 2 to get detailed update messages |

To implement beam current modulation, have your source’s NumberOfHistoriesInRun get its value from a time feature, as in:

| Source/MySource/NumberOfHistoriesInRun = Tf/MyBCMTimeFeature/Value |

By default, scorers will output just once, after the entire session. But if you wish to have separate output from specific runs:

| Sc/MyScorer/OutputAfterRun = "True"  # set True to trigger output of scorer after this run |

- If this is False, or not defined, we just output at the end of the simulation.
- If this is True, we output after every run.

### Random Time Mode

Random Time Mode generates one history per run, with a randomly sampled time at each run. This has several uses:

- It allows one to sample time in a continuous fashion, so may show features that are obscured by Sequential Mode
- It provides a way to do a lower statistics run of what would have been a very long Sequential Mode job, yet still see aspects of the entire time interval, rather than just the first subset of the sequential times

To run in Random Time Mode, specify the TimelineStart and TimelineEnd, turn on RandomizeTimeDistribution, and set your source’s NumberOfHistoriesInRandomJob, as in:

| Tf/RandomizeTimeDistribution = "True"  # defaults to "False" |
| Tf/TimelineStart = 0. s  # defaults to zero |
| Tf/TimelineEnd = 10. s  # must be larger than TimelineStart |
| Source/MySource/NumberOfHistoriesInRandomJob = 1000  # defaults to 100 |
For each history, a random time will be sampled between *TimelineStart* and *TimelineEnd*.

We keep the parameters that control random mode (*NumberOfHistoriesInRandomJob*) separate from those that control sequential mode (*NumberOfHistoriesInRun* and *NumberOfSequentialTimes*) so that you can easily switch between the two modes (by just switching *RandomizeTimeDistribution*).

To implement beam current modulation, give your source a time-dependent *ProbabilityOfUsingAGivenRandomTime*, as in:

```
d:So/MySource/ProbabilityOfUsingAGivenRandomTime = Tf/MyBCMTimeFeature/Value
```

### Fixed Time but with Very Large Number of Histories

The maximum number of histories possible per run is limited by the size of some of Geant4’s internal counters. If you need more than $10^9$ histories at a fixed time, you can work around this limitation by breaking your session into multiple runs:

- Set *Tf/NumberOfSequentialTimes* to some value greater than 1
- No need to actually set *TimelineStart* or *TimelineEnd* (they both default to 0)

Your total number of histories will then be $\text{NumberOfSequentialTimes} \times \text{NumberOfHistoriesInRun}$.

### Multithreading

TOPAS fully supports the Multi-Threaded simulation capability of Geant4.

By default, TOPAS will occupy just one CPU thread. To use more, adjust:

```
i:Ts/NumberOfThreads = 4  # defaults to 1
```

- If set to a positive integer, TOPAS will use that number of threads
- If set to 0, TOPAS will use all of your computer’s threads. This may be number of hardware cores or number of virtual cores (which includes hyper-threading cores) depending on your hardware architecture.
- If set to a negative number, TOPAS will use all BUT this number of threads, leaving you some threads reserved for other tasks (email, web browsing, etc.).

By default, console output from various threads will be interleaved. Output from each worker thread will have a distinctive prefix, such as:

```
G4WT0 >
G4WT1 >
```

To instead make Geant4 buffer the output, showing first everything from one thread and then everything from another thread, set:

```
b:Ts/BufferThreadOutput = "True"  # Causes console output to be show one thread at a time
```

### Random Number Seed

To set the random seed:
To generate several statistically independent runs, give each run a different `Ts/Seed`. A typical solution to produce 10 independent runs would be to give starting seeds of 1 to 10. The allowed range is 0 to 2147483647 (the maximum 32 bit integer).

For more details see the discussion in the Geant4 Application Developer’s Guide.

We use the random engine called `RanecuEngine` and the seed given to TOPAS is passed to the engine through `CLHEP::HepRandom::setTheSeed`.

**Miscellaneous**

### Interactive Geant4 Sessions

To have TOPAS pause and wait for interactive Geant4 commands:

```plaintext
b:Ts/PauseBeforeInit = "True"
b:Ts/PauseBeforeSequence = "True"
b:Ts/PauseBeforeQuit = "True"
```

After each pause, type the Geant4 command `exit` to return control to TOPAS.

- Most users will only use `PauseBeforeQuit`, typically to make a graphics window stay open at the end of the session (graphics windows close when Geant4 quits).
- The other two options, `PauseBeforeInit` and `PauseBeforeSequence`, provide the ability to enter Geant4 commands by hand, which may be useful in certain tests, but invalidates the basic TOPAS concept that the behavior of your simulation should be perfectly defined by TOPAS parameters.

### Dump Parameter Values

Additional overall control parameters are:

```plaintext
b:Ts/DumpParameters = "True" # dump full list of params to file TopasParameterDump_Run0.html
b:Ts/DumpNonDefaultParameters = "False" # Like above but omits defaults
sv:Ts/DumpParametersToSimpleFile = 2 "SomeParameter" "SomeOtherParameter" # Dumps the requested parameter types, names and values to a simple, human-readable file, TopasParameterDump_Run0.txt
sv:Ts/DumpParametersToSemicolonSeparatedFile = 2 "SomeParameter" "SomeOtherParameter" # Dumps the requested parameter types, names and values to a semicolon separated file, TopasParameterDumpSSF_Run0.txt. This file is suitable for easy import into other applications
```

### Verbosity

Additional overall control parameters are:

```plaintext
i:Ts/ShowHistoryCountAtInterval = 1 # how often to print history count to the console
# If set to 0, history count will never be printed
```
Other

Additional overall control parameters are:

- \texttt{b:Ts/ShowHistoryCountOnSingleLine = "False"} # Make count reuse a single line of console.
- \texttt{i:Ts/TrackingVerbosity = 0} # Set to larger integer to see details of tracking.

Quick Ways to Deactivate Parts of the Parameters Files

For most parameter categories, there is one key kind of parameter that triggers creation:

<table>
<thead>
<tr>
<th>Parameter object type</th>
<th>Triggering pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>\texttt{El/.../Symbol =}</td>
</tr>
<tr>
<td>Material</td>
<td>\texttt{Ma/.../Components =}</td>
</tr>
<tr>
<td>Component</td>
<td>\texttt{Ge/.../Parent =}</td>
</tr>
<tr>
<td>Particle Source</td>
<td>\texttt{So/.../Type =}</td>
</tr>
<tr>
<td>Physics setup</td>
<td>\texttt{Ph/.../Type =}</td>
</tr>
<tr>
<td>Scorer</td>
<td>\texttt{Sc/.../Quantity =}</td>
</tr>
<tr>
<td>Graphic View</td>
<td>\texttt{Gr/.../Type =}</td>
</tr>
<tr>
<td>Variance Reduction setup</td>
<td>\texttt{Vr/.../Type =}</td>
</tr>
<tr>
<td>Time Feature</td>
<td>\texttt{Tf/.../Function =}</td>
</tr>
</tbody>
</table>

Thus you could effectively comment out the entire Component, Element, Material, Particle Source, etc. by just commenting out that line. But this way of turning something off can get you into trouble since you may then inherit behavior from a parent parameter file.

A better way to handle this is by setting a specific parameter designed for this purpose:

- \texttt{Ge/MyComponent/Include = "False"}
- \texttt{So/MySource/NumberOfHistoriesInRun = 0}
- \texttt{Sc/MyScorer/Active = "False"}
- \texttt{Gr/MyGraphics/Active = "False"}

Such a parameter can then even be controlled by a time feature.
We have pre-defined a few materials. You are free to define additional materials, as in:

```
sv:Ma/Air/Components=4 "Carbon" "Nitrogen" "Oxygen" "Argon"  # names of elements
uv:Ma/Air/Fractions=4 0.000124 0.755268 0.231781 0.012827  # fractions of elements
d:Ma/Air/Density=1.2048 mg/cm³
s:Ma/Air/MeanExcitationEnergy=85.7 eV
s:Ma/Air/DefaultColor="lightblue"
```

All Elements have been pre-defined with natural isotope abundance from the NIST database. You will only need to create your own Elements if you need something other than natural isotope abundance. For that, see Elements and Isotopes below.

Fractions are by weight.

MeanExcitationEnergy is the $I$ parameter in the Bethe equation, which not only includes ionization, but also inner-atomic excitations, etc.

In the Default Parameters section, we show the complete list of pre-defined materials. This basically covers those materials that are used in our included examples.

You may also use any of the Materials and Compounds that are defined by default in Geant4. The names start with the prefix, G4_, such as: G4_A1, G4_T1, G4_MUSCLE_SKELETAL_ICRP, etc. The complete list of these materials and compounds can be found here.

- NIST material names must be specified with exact case.
- As of this writing, the mean excitation energy listed in the above reference for G4_WATER is incorrect. It lists G4_WATER mean excitation energy as 75.0 eV but it is actually set to 78.0 eV.

Note: The Geant4-DNA physics processes have special behavior for G4_WATER. They take into account the material’s molecular properties rather than just the atomic properties. Accordingly, you should use G4_WATER rather than defining your own Water, unless you have some other reason to make a specific change (such as changing the mean excitation energy to something other than 78.0 eV).
TOPAS Documentation, Release 3.0

It is up to you to define any additional materials that you want in your own parameter files. If you make your own material, make sure to pick a new material name (the string after the Ma/) and make sure that any other parameter file that uses this material includes the file where you defined this material (either directly or through Parameter File Chains). The usual rules of Parameter File Graphs govern parameter conflicts.

**Warning:** Do not use the prefix G4_ for the materials that you add. This prefix is reserved for materials and compounds from the pre-defined NIST database.

Where a pre-defined material definition exists, it is generally better to use that definition rather than making your own material. The pre-defined material may provide extra benefit by triggering specific corrections to ionization models.

If you have a set of materials that differ only in density, you can define them all at once (this is commonly needed for imaging to material conversion):

```plaintext
i:Ma/MyMaterial/VariableDensityBins = 100
u:Ma/MyMaterial/VariableDensityMin = .1
u:Ma/MyMaterial/VariableDensityMax = 10.
```

will generate 100 versions of MyMaterial, with densities varying from .1 x normal to 10. x normal. The material names will then be like:

```plaintext
MyMaterial_VariableDensityBin_0
MyMaterial_VariableDensityBin_1
...
MyMaterial_VariableDensityBin_99
```

**Elements and Isotopes**

All Elements have been pre-defined with natural isotope abundance from the NIST database. You will only need to create your own Elements if you need something other than natural Isotope abundance. You can define additional elements as follows:

Define each isotope that you will use, specifying Z, N and A:

```plaintext
i:Is/U235/Z = 92
i:Is/U235/N = 235
d:Is/U235/A = 235.01 g/mole
i:Is/U238/Z = 92
i:Is/U238/N = 238
d:Is/U238/A = 238.03 g/mole
```

Define your element with your desired proportion of these isotopes:

```plaintext
g:El/MyEIU/Symbol = "MyEIU"
sv:El/MyEIU/IsotopeNames = 2 "U235" "U238"
uv:El/MyEIU/IsotopeAbundances = 2 90. 10.
```

See Isotope.txt example.
CHAPTER 11

Geometry Components

Introduction

All Geometry Components must have at least the following parameters:

```plaintext
s:Ge/MyComponent/Parent = "World"
s:Ge/MyComponent/Type = "TsBox"
d:Ge/MyComponent/TransX=0.0 cm # defaults to 0
d:Ge/MyComponent/TransY=0.0 cm # defaults to 0
d:Ge/MyComponent/TransZ=0.0 cm # defaults to 0
d:Ge/MyComponent/RotX=0.0 deg # defaults to 0
d:Ge/MyComponent/RotY=0.0 deg # defaults to 0
d:Ge/MyComponent/RotZ=0.0 deg # defaults to 0
```

The `Parent`, `Trans` and `Rot` parameters place a component within its “mother” as described in *Placement of Components*.

Each `Type` has its own set of additional required parameters, discussed elsewhere for each specific component type.

The component name can include the forward slash character `/`, and this is used in many examples to give some hints about component hierarchy, such as:

```plaintext
s:Ge/VBox2/Dipole/Parent = "Nozzle"
```

This bit of hierarchy in the component name, such as `VBox2/Dipole`, does NOT actually control how the components are assembled. The actual control is from the `Parent` parameter (discussed here). The forward slash is just another character here. You could just as well use `VBox2_Dipole` or `VBox2Dipole`, as long as you use the same exact string whenever you refer to this component.

Components that are in the real world (as opposed to *Parallel Worlds*) must also have a material:

```plaintext
s:Ge/MyComponent/Material = "Air"
```

To deactivate a Component (and all its children), you can either comment out the parameter that sets its `Parent`, or set its `Include` parameter to false, as in:

```plaintext
s:Ge/MyComponent/Include = false
```
While it is not forbidden to have unused components (components that are never assigned a Parent), this can often be a sign that you have not correctly assigned the parents in your geometry. Accordingly, we check for unused components on startup and given a warning message if any are found. You can disable this warning message by setting:

```
Ge/CheckForUnusedComponents = "False"
```

In some cases you may want to keep unused components around. This can be like keeping extra pieces of unused laboratory equipment handy on a shelf. They will have no effect on your simulation, but remain available to quickly plug in when needed by assigning a parent and setting placement parameters.

Physics control for a specific component is done as part of the Ge/ parameters for that component rather than in the Ph/ parameters, such as:

```
d:Ge/MyComponent/MaxStepSize = 1. mm # sets maximum step size used in this component
```

### Placement of Components

A component’s Parent parameter tells which other component the current one is a child of. In this way, one can build a hierarchy of components:

```
s:Ge/MyComponent/Parent = SomeOtherComponent
```

The one component that is always provided automatically for you, into which you plug the rest of your hierarchy, is called World.

Each component has three translation and three rotation parameters. These give the position of the component in the coordinate system of its parent component.

The following defines a box of air with half width of 5 m on each side placed at the center of the world:

```
s:Ge/MyBox/Type="TsBox"
s:Ge/MyBox/Parent = "World"
s:Ge/MyBox/Material="Air"
d:Ge/MyBox/HLX=5. m # Half Length
d:Ge/MyBox/HLY=5. m
d:Ge/MyBox/HLZ=5. m
d:Ge/MyBox/TransX=0. m
d:Ge/MyBox/TransY=0. m
d:Ge/MyBox/TransZ=0. m
d:Ge/MyBox/RotX=0. deg
d:Ge/MyBox/RotY=0. deg
d:Ge/MyBox/RotZ=0. deg
```

If you set more than one rotation, note that rotation happens first around X, then the Y rotation is applied around the now-rotated axes, and then the Z rotation is applied around those rotated axes. In general, a way to keep rotations more clear is to use intermediate Group components as follows:

- Place your component inside a Group component.
- Rotate your component around one axis.
- Rotate the group component around the other axis.
While the direction of rotation can be confusing, the correctness of rotations in TOPAS has been double, triple and quadruple checked and found consistent with the definitions in Geant4. The `Rotation.txt` example shows an object being rotated first in the positive X direction, then in the positive Y direction, then in the positive Z direction.

For Geant4 experts, be advised that the rotation angles you provide to TOPAS are passed into `G4RotationMatrix()->rotateX/Y/Z`. Further discussion of Geant4 rotations can be found here.

The following overrides the size definition of the `World` box that was inherited from the built-in default parameters and then inserts a box into this world and another box into the first box:

```
# Overrides the world size that was set in built-in defaults:
Ge/World/HLX=10. m
Ge/World/HLY=10. m
Ge/World/HLZ=10. m

# Box inserted into the World
s:Ge/TestBox/Material="Air"
s:Ge/TestBox/Parent="World"
s:Ge/TestBox/Type="TsBox"
d:Ge/TestBox/HLX=400. cm
d:Ge/TestBox/HLY=300. cm
d:Ge/TestBox/HLZ=200. cm
d:Ge/TestBox/TransX=0. m
d:Ge/TestBox/TransY=0. cm
d:Ge/TestBox/TransZ=0. m
d:Ge/TestBox/RotX=0. deg
d:Ge/TestBox/RotY=0. deg
d:Ge/TestBox/RotZ=0. deg

# Another box inserted into the first box
s:Ge/TestBox2/Material="Carbon"
s:Ge/TestBox2/Parent="TestBox"
s:Ge/TestBox2/Type="TsBox"
d:Ge/TestBox2/HLX=180. cm
d:Ge/TestBox2/HLY=120. cm
d:Ge/TestBox2/HLZ=80. cm
d:Ge/TestBox2/TransX=0. m
d:Ge/TestBox2/TransY=0. cm
d:Ge/TestBox2/TransZ=150. cm
d:Ge/TestBox2/RotX=0. deg
d:Ge/TestBox2/RotY=30. deg
d:Ge/TestBox2/RotZ=0. deg
```

**Overlap Checking**

Because accidental overlaps of geometry volumes are a serious issue for all Monte Carlo simulations, Geant4 provides tools to automatically check for such overlaps. Overlap checking is not perfect; it works by testing a random set of points on each boundary, to see if they are inside any other boundary. Thus it will not necessarily find all overlaps. By default Geant4 checks 100 points on each volume (we will eventually provide a command to adjust this number of points). Overlap checking has a speed cost at initialization time, so once you are confident that your geometry has no overlaps, you may want to turn this feature off:

```
Ge/CheckForOverlaps = "False"
```
Pre-Defining Values

It may be useful to pre-define a range of named-values, such that you can easily access the values later. For example, we pre-define the angles at which certain scatterers are stored on a scatterer selection wheel:

```
d:Ge/Gantry1/Scatterer2/RotZForSS0 = 0. deg  
d:Ge/Gantry1/Scatterer2/RotZForSS8 = 270. deg  
d:Ge/Gantry1/Scatterer2/RotZForSS2 = 180. deg  
d:Ge/Gantry1/Scatterer2/RotZForSS3 = 90. deg
```

And then in our user file, the user doesn’t have to know these actual angles, but can just rotate to one of the named scatterers:

```
Ge/Scatterer2/Holder/RotZ = Ge/Gantry1/Scatterer2/RotZForSS3 deg
```

Parallel Worlds

Components can be assigned to “parallel worlds” rather than the standard, mass world. Such components have no effect on physics (other than usually very minor step limitation effects) but can still be used for scoring. Such components can arbitrarily overlay the mass world. Their volumes can overlap any other volumes in other mass or parallel worlds. To assign a component to a parallel world, include the line:

```
b:Ge/MyComponent/IsParallel = "True"
```

- Parallel world components may be children of other parallel world components.
- Parallel world components may be children of mass world components.
- Mass world components may not be children of parallel world components.

A new parallel world will be created each time you specify `IsParallel`, with an automatically generated parallel world name based on the component name. If you want to explicitly assign multiple components to the same parallel world, provide the additional parameter:

```
s:Ge/MyComponent/ParallelWorldName = "SomeParallelWorldName"
```

This is useful if you will have many components in parallel worlds, as Geant4 only allows up to 8 total worlds.

**Warning:** In certain cases, TOPAS must represent a geometry by using a Geant4 technique called “parameterized volumes.” However we have found that Geant4 behaves unreliably if parameterized sphere is placed in a parallel world. Accordingly, TOPAS applies a safety restriction:

- TsSphere can not be in a parallel world if it has any divisions.

Layered Mass Geometry

Components that are in a parallel world can have material or not. If they have material, and they are listed in the `LayeredMassGeometryWorlds` parameter, this material will take precedence over any real world material found in that location.

In Geant4 this is called Layered Mass Geometry. It is further described in (PubMed):

Any time a component in a parallel world has material, that world must be listed in the `LayeredMassGeometryWorlds` parameter. The parameter is a string vector because any number of parallel worlds can have material. The order of the worlds in this parameter is significant. Material from worlds listed later in this list take precedence over material in worlds listed earlier. Thus, in the following example, material in the world `Seed` will take precedence over material in the world `Tumor` which will take precedence over material in the regular world:

```plaintext
sv:Ph/Default/LayeredMassGeometryWorlds = 2 "Tumor" "Seed"
```

A simple example is provided in `LayeredMassGeometry.txt`.

### Electromagnetic Fields

You can assign an electric, magnetic or combined electromagnetic field to any geometry component (with exception of Group components, which have no intrinsic extent). The field will extend into any child components unless they themselves have their own field.

To assign a field, add the parameter `Field`, as in:

```plaintext
s:Ge/MyComponent/Field = "DipoleMagnet" # "DipoleMagnet", "QuadrupoleMagnet", "MappedMagnet", "UniformElectroMagnetic" or your own definition
```

For "DipoleMagnet", specify dipole field and strength, as in (see `DipoleMagnet.txt`):

```plaintext
u:Ge/MyComponent/MagneticFieldDirectionX = 0.0
u:Ge/MyComponent/MagneticFieldDirectionY = 1.0
u:Ge/MyComponent/MagneticFieldDirectionZ = 0.0
d:Ge/MyComponent/MagneticFieldStrength = 3.0 tesla
```

For "QuadrupoleMagnet", specify the two components of the gradient, as in (see `QuadrupoleMagnet.txt`):

```plaintext
d:Ge/MyComponent/MagneticFieldGradientX = 1.0 tesla
d:Ge/MyComponent/MagneticFieldGradientY = 1.0 tesla
```

For "MappedMagnet", specify a field map in the Opera 3D format, as in (see `PurgingMagnet_move.txt`):

```plaintext
s:Ge/MyComponent/MagneticField3DTable = "PurgMag3D.TABLE"
```

For "UniformElectroMagnetic", specify electric field and dipole magnetic field, as in (see `UniformElectroMagneticField.txt`):

```plaintext
u:Ge/MyComponent/ElectricFieldDirectionX = 1.0
u:Ge/MyComponent/ElectricFieldDirectionY = 1.0
u:Ge/MyComponent/ElectricFieldDirectionZ = 0.0
d:Ge/MyComponent/ElectricFieldStrength = 5000 kV/cm
u:Ge/MyComponent/MagneticFieldDirectionX = 0.0
u:Ge/MyComponent/MagneticFieldDirectionY = 1.0
u:Ge/MyComponent/MagneticFieldDirectionZ = 0.0
d:Ge/MyComponent/MagneticFieldStrength = 5.0 tesla
```

If you have any other value in `Field`, TOPAS will look in your extensions to find your own class that defines this field. See `Custom Fields` for details on writing extension fields.

Field orientation is set by rotating the component.
As with almost any TOPAS parameter, the Electric Field Strength, Dipole Magnet Strength, Quadrupole Magnet Gradient or Mapped Magnetic Field file can be set to change over time by using Time Features such as (see QuadAndDipoleMagnets.txt):

```plaintext
d:Ge/MyComponent/MagneticFieldStrength = Tf/BField1st/Value tesla
```

Fine control of the stepping algorithm can be done by changing the following parameters from their default values:

```plaintext
s:Ge/MyComponent/FieldStepper = "ClassicalRK4"
d:Ge/MyComponent/FieldStepMinimum = 1.0 mm
d:Ge/MyComponent/FieldDeltaChord = 1.0e-1 mm
```

See the Geant4 Application Developers Guide for detailed discussion of these options.

Stepper choices for purely magnetic fields are:

- “ExplicitEuler”
- “ImplicitEuler”
- “SimpleRunge”
- “SimpleHeum”
- “HelixExplicitEuler”
- “HelixImplicitEuler”
- “HelixSimpleRunge”
- “CashKarpRKF45”
- “RKG3”
- “ClassicalRK4”

Stepper choices for electromagnetic fields are:

- “ExplicitEuler”
- “ImplicitEuler”
- “SimpleRunge”
- “SimpleHeum”
- “ClassicalRK4”

**Visualization Attributes**

By default, Components are colored according to their materials, based on the default color that was assigned to the material, such as:

```plaintext
s:Ma/Air/DefaultColor="lightblue"
```

Parameters of the Component let you override this color and set other visualization attributes:

```plaintext
s:Ge/MyComponent/Color = "red"
s:Ge/MyComponent/DrawingStyle = "Solid" # "Solid", "Wireframe" or "FullWireFrame"
i:Ge/MyComponent/VisSegsPerCircle = 100 # Number of line segments to use to approximate a circle, defaults to 24
b:Ge/MyComponent/Invisible = "True" # defaults to False meaning visible
```
FullWireFrame includes drawing of additional edge lines that Geant4 calls “soft edges” - on many graphics devices WireFrame and FullWireFrame give the same result.

Increase VisSegsPerCircle if you want a smoother curve.

### Dividable Components

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
<th>Type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TsBox</td>
<td>XBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>YBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ZBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td>TsCylinder</td>
<td>RBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>PhiBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ZBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td>TsSphere</td>
<td>RBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>PhiBins</td>
<td>i</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>ThetaBins</td>
<td>i</td>
<td>1</td>
</tr>
</tbody>
</table>

Scorers associated with the dividable components may use the same or different divisions (thus one can do things like represent the patient with CT resolution but score with other resolutions). See here for details.

You cannot place child components inside a divided component, but if the only reason for dividing this component is to have fine-grained scoring, you can easily work around this limitation. Use an undivided parent component. Place the children into this undivided parent component. Then when you specify that you want to score on this parent
component, specify divided scoring (see here). TOPAS will automatically create a parallel world version of your component to handle the divided scoring.

You can optionally specify different materials for each voxel, overriding the value set in the regular /Material parameter:

```
sv:Ge/Phantom/VoxelMaterials = 100 "G4_WATER" "G4_WATER" "Air" "Air" "G4_WATER" ...
```

This means you can create complex phantoms directly from the parameter system. VoxelMaterials works for all three kinds of divided components: TsBox, TsCylinder and TsSphere. See the DoseInVoxelMaterials.txt example.

**Generic Components**

You can create a Geometry Component for any of the standard solids defined in the geometry section of the Geant4 Application Developers Guide.

The ShapeTestWithAllParameters.txt example demonstrates how to build each of the solids.

Below we list the parameters for each Geant4 solid. Further details about the parameters, along with helpful diagrams, can be found in the Geant4 Application Developers Guide. For most solids, sizes are specified in Half Lengths, denoted with an HL, such as HLX. For a few solids, sizes are specified in full Lengths, denoted with just L, such as LX.

**Note:** The TsBox, TsCylinder and TsSphere should be used instead of G4Box, G4Tubs and G4Sphere respectively. See Dividable Components.

Some examples of components that can be built just from Generic Components:

- Scatterer
- Collimator
- Mirror
- Water Tank
- Rando Phantom (as constructive solid geometry rather than DICOM import)
- Pin Diode Chamber
- Flat Panel Imaging Device
- Standard Ion Chamber
- Segmented Ion Chamber
- Faraday Cup

We have built some complex things just from combinations of the Generic Components (such as the STAR radiosurgery beamline at MGH).
<table>
<thead>
<tr>
<th>Type</th>
<th>Parameters</th>
<th>Type</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>G4CutTubs</td>
<td>RMin, RMax, HL, SPhi, DPhi, LowNorm, HighNorm</td>
<td>d, d, d, d, uv (3), uv (3)</td>
<td>0 cm, 0 deg, 360 deg</td>
</tr>
<tr>
<td>G4Cons</td>
<td>RMin1, RMax1, RMin2, RMax2, HL, SPhi, DPhi</td>
<td>d, d, d, d, d, d</td>
<td>0 cm, 0 cm, 0 deg, 360 deg</td>
</tr>
<tr>
<td>G4Para</td>
<td>HLX, HLY, HLZ, Alpha, Theta, Phi</td>
<td>d, d, d, d, d, d</td>
<td></td>
</tr>
<tr>
<td>G4Trd</td>
<td>HLX1, HLX2, HLY1, HLY2, HLZ</td>
<td>d, d, d, d, d</td>
<td></td>
</tr>
<tr>
<td>G4RTrap</td>
<td>LZ, LY, LX, LTX</td>
<td>d, d, d, d</td>
<td></td>
</tr>
<tr>
<td>G4GTrap</td>
<td>HLZ, Theta, Phi, HLY1, HLX1, HLX2, Alp1, HLY2, HLX3, HLX4, Alp2</td>
<td>d, d, d, d, d, d, d, d, d, d</td>
<td></td>
</tr>
</tbody>
</table>

11.7. Generic Components
**Group Component**

Creates no actual solid, but still has a placement (Trans and Rot). Other components placed within this Group component are affected by this placement just as if the group were an enclosing box component.

The following defines a group component called `MyGroup`:

```plaintext
s:Ge/MyGroup/Type="Group"
s:Ge/MyGroup/Parent = "World"
d:Ge/MyGroup/TransX=2. m
d:Ge/MyGroup/TransY=2. m
d:Ge/MyGroup/TransZ=0. m
d:Ge/MyGroup/RotX=0. deg
d:Ge/MyGroup/RotY=0. deg
d:Ge/MyGroup/RotZ=30. deg
```

The following example shows how a Group Component, `Jaws`, placed in a nozzle, allows one to position two individual movable collimator blocks, `Jaw_Upper` and `Jaw_Lower`, without the creation of an extraneous mother volume:

```plaintext
s:Ge/Jaws/Type = "Group"
s:Ge/Jaws/Parent = "Nozzle"
d:Ge/Jaws/TransZ = 0. m ... 
s:Ge/Jaw_Upper/Type = "TsBox"
s:Ge/Jaw_Upper/Parent = "Jaws"
s:Ge/Jaw_Upper/Material = "Tungsten"
d:Ge/Aperture/TransY = 2. cm ...
... 
s:Ge/Jaw_Lower/Type = "TsBox"
s:Ge/Jaw_Lower/Parent = "Jaws"
s:Ge/Jaw_Lower/Material = "Tungsten"
d:Ge/Aperture/TransY = -2. cm ...
```

**Specialized Components**

<table>
<thead>
<tr>
<th>Geometry Component</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range Modulator Wheel</td>
<td>TsRangeModulator</td>
</tr>
<tr>
<td>Propeller</td>
<td>TsPropeller</td>
</tr>
<tr>
<td>Ridge Filter</td>
<td>TsRidgeFilter</td>
</tr>
<tr>
<td>Multi Wire Chamber</td>
<td>TsMultiWireChamber</td>
</tr>
<tr>
<td>Multi Leaf Collimator</td>
<td>TsMultiLeafCollimator</td>
</tr>
<tr>
<td>CAD (Computer Aided Design)</td>
<td>TsCAD</td>
</tr>
<tr>
<td>Aperture</td>
<td>TsAperture</td>
</tr>
<tr>
<td>Compensator</td>
<td>TsCompensator</td>
</tr>
</tbody>
</table>

Each of the specialized components has its own set of special parameters. Usage is best learned by studying the relevant examples parameter files included in TOPAS.

You may write your own additional components (see Custom Geometry Components).

The following figure from Samsung Medical Center shows how their very specific quadrupole magnet system was coded as a TOPAS geometry.
Range Modulator Wheel

TOPAS Range modulator is designed to accommodate various specifications from a vendor. We suggest modeling your Range Modulator Wheel (RMW) by the following procedure:

- Define the dimension of RMW drum, such as thickness and material of shell and hub (see figure below). Tracks will be placed in between the hub and the shell.

- This space (in between hub and shell) is vertically divided into three sections named, “Upper”, “Middle”, and “Bottom” so that each section can have its own tracks. You can adjust heights of these sections. The sum of these heights is the total height of your RMW.

- In order to reserve spaces for tracks, divide radially each section into as many as tracks you want by using the parameter, RadialDivision.

- Using vector parameters, configure the tracks individually such as each block’s height, span angle, and material. Then assign vector parameter to the parameter, called Pattern.

Illustration for TOPAS RMW dimensions. Tracks are placed in between Rout of Hub and Rin of Shell and this area is to be radially divided in case of placing multiple tracks. There are three vertical rooms, so it is possible to make double sided RMWs with an interface disk.
An example of RMW; (a) Perspective view. Upper section is divided into two but only inner radial division has a track pattern. In middle section, two track patterns are used to make a hole. (b) X-Y view from +z of RMW and (c) X-Y view from -z of RMW. Tracks are drawn in wireframe style, so more lines on the tracks are shown than the number of blocks.

Here is the complete set of the parameters for the above RMW (see RangeModulator.txt example):

```
# Common parameters: type of geometry, position, and rotation
s:Ge/RangeModulatorA/Type = "TsRangeModulator"
s:Ge/RangeModulatorA/Material = "Parent"
s:Ge/RangeModulatorA/Parent = "World"
d:Ge/RangeModulatorA/TransX = 10.0 cm
d:Ge/RangeModulatorA/TransY = 0.0 cm
d:Ge/RangeModulatorA/TransZ = 0.0 cm
d:Ge/RangeModulatorA/RotX = 0.0 deg
d:Ge/RangeModulatorA/RotY = 0.0 deg
d:Ge/RangeModulatorA/RotZ = 0.0 deg
b:Ge/RangeModulatorA/Invisible = "TRUE"

# Set height of each sections and total height = 160.0 mm
d:Ge/RangeModulatorA/HeightOfUpper = 150 mm
d:Ge/RangeModulatorA/HeightOfMiddle = 1.0 mm
d:Ge/RangeModulatorA/HeightOfLower = 9.0 mm

# Shell dimensions, material, color, etc.
d:Ge/RangeModulatorA/Shell/Rin = 15.0 cm
d:Ge/RangeModulatorA/Shell/Rout = 15.5 cm
s:Ge/RangeModulatorA/Shell/Material = "Aluminum"
s:Ge/RangeModulatorA/Shell/Color = "grey"
```
In the same way, you can configure other tracks. Then the track1 on upper area looks like following figure.
A track pattern from the parameter above; (left) a complete set of the track view. (right) blocks are constructed in counterclockwise.

When TOPAS builds the geometry, you will see the numbers are input properly from console output as:

```
---UpperTrack1 , # of Blocks: 14
  0th Block
    Angle : 5, 115 deg
    Height : 7.7 cm
    Material: Ts_Lexan
  1st Block
    Angle : 115, 146.5 deg
    Height : 8.2 cm
    Material: Ts_Lexan
  2nd Block
    Angle : 146.5, 173.2 deg
    Height : 8.7 cm
    Material: Ts_Lexan
  3rd Block
    Angle : 173.2, 195.07 deg
    Height : 9.215 cm
    Material: Ts_Lexan
  4th Block
    Angle : 195.07, 216.15 deg
    Height : 9.5 cm
    Material: Ts_Lexan
...
```

TOPAS RMW is a specialized geometry and so allows only the rotation around z-axis as well as the propeller rotation. Two examples demonstrate how to rotate RMW and modulate beam current using *Time Features* (`RangeModulator_ConstantBeam.txt` and `RangeModulator_CurrentModulatedBeam.txt`).
Propeller

A propeller is a component widely used to modulate the range of Bragg peaks. TOPAS currently supports a symmetrical propeller, i.e., each blade has the same shape but in different placements. Users can specify the number of blades with a spanning angle, thickness and materials of each layer. Here is an example of a single-layer propeller having 4 blades.

Each blade is constructed in the counterclockwise order. The figure shows its shape with coordination system.

Here is the complete set of the parameters for the above Propeller:

```sh
# Common parameters: type of geometry, position, and rotation
s:Ge/PropellerA/Type = "TsPropeller" #TsPropeller as type of geometry
s:Ge/PropellerA/Parent = "World"
# This is required to be set as "Parent"

s:Ge/PropellerA/TransX = 0.0 cm
s:Ge/PropellerA/TransY = 0.0 cm
s:Ge/PropellerA/TransZ = 0.0 cm
s:Ge/PropellerA/RotX = 0.0 deg
s:Ge/PropellerA/RotY = 0.0 deg
s:Ge/PropellerA/RotZ = 0.0 deg

b:Ge/PropellerA/Invisible = "true" #To avoid visualize propeller’s mother volume.
```
i:Ge/PropellerA/NbOfBlades = 4 #Number of blades
d:Ge/PropellerA/Rin = 10.0 mm #Inner radius, Rin in the figure
d:Ge/PropellerA/Rout = 127.5 mm #Outer radius, Rout in the figure
dv:Ge/PropellerA/Thickness = 1 0.356 mm #thickness.
dv:Ge/PropellerA/Angles = 1 63.15 deg #span angle of each blade.
sv:Ge/PropellerA/Materials = 1 "G4_POLYVINYL_ACETATE" #material.

# Visualization attributes setting
g:Ge/PropellerA/Blade/Material = Ge/PropellerA/Material
g:Ge/PropellerA/Blade/Color = "skyblue"
g:Ge/PropellerA/Blade/DrawingStyle = "Solid"
i:Ge/PropellerA/Blade/VisSegsPerCircle = 360
# you can apply a propeller specific step size
d:Ge/PropellerA/Blade/MaxStepSize = 10 mm
b:Ge/PropellerA/PrintInformation = "True" #Print out specification, see below

While TOPAS starts to build geometries, you can confirm whether the numbers you put are input properly from console output as:

Layer: "0" , Thickness: 0.0356 (cm), Angle: 63.15 (deg), Material: G4_POLYVINYL_ACETATE
Blade "0", Angle (-31.575 deg, 31.575deg)
Blade "1", Angle (88.425 deg, 151.575deg)
Blade "2", Angle (208.425 deg, 271.575deg)

With different numbers of blades, the angle of each blade will look like:

(Left) Ge/PropellerA/NbOfBlades = 2, (Right) Ge/PropellerA/NbOfBlades = 3.

You can model a multiple layered propeller just by extending the vector parameters, such as Thickness, Angles, and Materials (for more detail, see Propeller.txt):

dv:Ge/PropellerA/Thickness=10
0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 mm
dv:Ge/PropellerA/Angles=10
63.15 54.15 46.450 40.65 35.85 31.8 28.1 24.725 21.8 19.1 deg
sv:Ge/PropellerA/Materials=10
"Lexan" "G4_WATER" "G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE"
Layers are created in the order of the parameter vector, i.e., Lexan is the bottom layer in this case, Water layer is the next, and so on. It is possible to make each layer with different thickness, angles, and materials. Note that these three vector parameters have same number of elements.

TOPAS propeller allows only the rotation around z-axis, which means that you can only assign rotation Time Feature to RotZ, such as Ge/PropellerA/RotZ = Tf/ContinuousRotation/Value. Two examples demonstrate how to handle propeller rotations (Propeller_ContinuousRotation.txt and Propeller_StepRotation.txt).

**Ridge Filter**

A ridge filter is an energy modulation component used in proton therapy. TOPAS offers a generic way to model an arbitrary shape of a ridge and place the replica. The shape of a ridge is defined in the x-z plane and then it becomes a volume by extending in the y direction.

(Left) A ridge shape in X-Z plane, represented by points-connection. Because the connection starts at the origin and ends at the last point, (width, 0), so users need to define the width of a ridge first. Depending on the topology of points, the arbitrary shape can be constructed. (Right) A complete ridge by extending the shape along with y axis.

Here is a complete set of the parameters for the above ridge filter (see RidgeFilter.txt example):
# Common parameters: type of geometry, position, and rotation

```plaintext
s:Ge/RidgeFilterA/Type = "TsRidgeFilter"
```

```plaintext
s:Ge/RidgeFilterA/Parent = "RidgeGroup"
```

```plaintext
s:Ge/RidgeFilterA/Material = "Aluminum"
```

```plaintext
d:Ge/RidgeFilterA/TransX = 0.0 cm
d:Ge/RidgeFilterA/TransY = 0.0 cm
d:Ge/RidgeFilterA/TransZ = 0.0 cm
d:Ge/RidgeFilterA/RotX = 0.0 deg
d:Ge/RidgeFilterA/RotY = 0.0 deg
d:Ge/RidgeFilterA/RotZ = 0.0 deg
```

```plaintext
s:Ge/RidgeFilterA/DrawingStyle = "Solid"
```

# Ridge Filter-specific parameters;
# Width definition
# Note that points are sequentially connected.
# Number of XPoints and YPoints should be same.

```plaintext
dv:Ge/RidgeFilterA/XPoints = 8
0.0 0.8 1.3 1.8 2.2 2.7 3.2 4.0 mm
dv:Ge/RidgeFilterA/YPoints = 8
2.4 4.0 9.1 14.0 14.0 9.1 4.0 2.4 mm
d:Ge/RidgeFilterA/Width = 4.0 mm
d:Ge/RidgeFilterA/Length = 1.0 cm
```

To check that the numbers are input properly:

```plaintext
b:Ge/RidgeFilterA/PrintInformation = "True"
```

will generate console output as:

```
Ridge points (x,z) --- :8
P initial : (0, 0) cm
P 0th : (0, 0.24) cm
P 1st : (0.08, 0.4) cm
P 2nd : (0.13, 0.91) cm
P 3rd : (0.18, 1.4) cm
P 4th : (0.22, 1.4) cm
P 5th : (0.27, 0.91) cm
P 6th : (0.32, 0.4) cm
P 7th : (0.4, 0.24) cm
P final : (0.4, 0) cm
```

You can make replicas of the ridge and their positions along the x-axis. A total of 3 replicas of the ridge and placed at -5.0, 0.0, 5.0. Each point represents the x-coordinate of the center of ridge width:

```plaintext
dv:Ge/RidgeFilterA/Displacement = 3 -5.0 0.0 5.0 mm
```

![Diagram of the Ridge Filter](image)
An example of replica set.

**Multi Wire Chamber**

A multi wire chamber may be built from many of geometry primitives such as TsBox and TsCylinder. However, it is quite cumbersome to place many wires individually and adjust their dimension on any request. So TOPAS multi wire chamber (TsMultiWireChamber) allows to instantiate many wires and to place them efficiently. TsMultiWireChamber is a box consisting of multiple sets of wires. Each set can have its own configuration, such as the dimension and material of the wires, spaces between wires, alignment axis, Z-positions, and drawing-style.

Here is an example of TsMultiWireChamber (see *MultiWire_Chamber.txt* example).

TOPAS multi wire chamber consists of two wire sets aligned along the X and Y axes. Three red wires are aligned to X axis while four gray wires are aligned to Y axis. These two sets of wires are placed within their mother box (gas filed).

The following parameters show how to model the above multi wire chamber:

```plaintext
s:Ge/WireChamberA/Parent = "World"
s:Ge/WireChamberA/Type = "TsMultiWireChamber" #Type of geometry
d:Ge/WireChamberA/HLX=30.0 cm #Chamber dimension
d:Ge/WireChamberA/HLY=30.0 cm
d:Ge/WireChamberA/HLZ=10.0 cm
s:Ge/WireChamberA/Material="Air" #Chamber is filled with this material.
d:Ge/WireChamberA/TransX=0.0 cm
d:Ge/WireChamberA/TransY=0.0 cm
d:Ge/WireChamberA/TransZ=0.0 cm
d:Ge/WireChamberA/RotX=0.0 deg
d:Ge/WireChamberA/RotY=0.0 deg
d:Ge/WireChamberA/RotZ=0.0 deg
i:Ge/WireChamberA/NbOfLayers=2 #Number of wire sets.

# Parameters for specifying each wire set add ‘Layer#’ to geometry name.
# So Layer1 to Layer’NumberOfLayers’.
d:Ge/WireChamberA/Layer1/RMin=0.0 cm #Wire’s inner radius
d:Ge/WireChamberA/Layer1/RMax=1.0 cm #Wire’s outer radius
s:Ge/WireChamberA/Layer1/Material="Brass" #Wire material
```

11.9. Specialized Components
When TOPAS starts to build geometries, you will see the numbers are input properly from console output as:

Layer: "0", # of Wires: 3, Alignment: X, Wire (Rmin= 0 cm, Rmax= 1 cm, HL= 20 cm),
- Z Position in the Chamber: 5 (cm)
  Wire "0", Position (0 cm, -10 cm)
  Wire "1", Position (0 cm, 0 cm)
  Wire "2", Position (0 cm, 10 cm)
Layer: "1", # of Wires: 5, Alignment: Y, Wire (Rmin= 0.5 cm, Rmax= 1 cm, HL= 20 cm),
- Z Position in the Chamber: -5 (cm)
  Wire "0", Position (-20 cm, 0 cm)
  Wire "1", Position (-10 cm, 0 cm)
  Wire "2", Position (0 cm, 0 cm)
  Wire "3", Position (10 cm, 0 cm)
  Wire "4", Position (20 cm, 0 cm)

Multi Leaf Collimator

Due to the design variations of Multi Leaf Collimator (MLC) from manufacturers, TOPAS provides a simplified MLC model instead of a generic design. With TOPAS MLC’s minimal set of parameters, users can define various width of each leaf and opening of each leaf.

Illustrations for TOPAS MLC dimensions. The user can define an arbitrary number of leaves with different width of each leaf. TOPAS detects leaf collision when it is built and leaves are repositioned by Time Features operations.
Here is a complete set of the parameters for the above TOPAS MLC (see `MultiLeafCollimator.txt` example):

```plaintext
# Common parameters: type of geometry, position, and rotation
s:Ge/MultiLeafCollimatorA/Type = "TsMultiLeafCollimator"
s:Ge/MultiLeafCollimatorA/Parent = "World"
s:Ge/MultiLeafCollimatorA/Material = "Aluminum"

# Ge/MultiLeafCollimatorA/TransX = 0.0 cm
# Ge/MultiLeafCollimatorA/TransY = 0.0 cm
# Ge/MultiLeafCollimatorA/TransZ = 0.0 cm

# Ge/MultiLeafCollimatorA/RotX = 0.0 deg
# Ge/MultiLeafCollimatorA/RotY = 0.0 deg
# Ge/MultiLeafCollimatorA/RotZ = 0.0 deg

s:Ge/MultiLeafCollimatorA/DrawingStyle = "Solid"
b:Ge/MultiLeafCollimatorA/PrintInformation = "True"

# MLC-specific parameters:
# Limits Leaf opening. Any of absolute values from X- Leaf can’t exceed this value.
d:Ge/MultiLeafCollimatorA/MaximumLeafOpen = 5.0 cm
d:Ge/MultiLeafCollimatorA/Thickness = 5.0 cm #Leaf thickness (z)
d:Ge/MultiLeafCollimatorA/Length = 6.0 cm #Leaf length (y)

dv:Ge/MultiLeafCollimatorA/Widths = 5 1.5 0.5 0.5 0.5 1.5 cm #Leaves width

# Each leaf’s opening distance from Y axis.
# XMminusLeavesOpen means the x position of X- leaf’s right edge.
# XPplusLeavesOpen means the x position of X+ leaf’s left edge.
dv:Ge/MultiLeafCollimatorA/XMinusLeavesOpen = 5 0.0 -0.3 -0.2 -0.5 0.0 cm
dv:Ge/MultiLeafCollimatorA/XPlusLeavesOpen = 5 0.0 0.3 0.2 0.5 0.0 cm

TOPAS MLC is a specialized geometry and so allows only the reposition of each leaf as a function of time, using Time Features (see `MultiLeafCollimator_sequence.txt` example).

**CAD (Computer Aided Design)**

The TsCAD component allows you to turn any geometry that has been designed in a CAD system into a TOPAS Component. This allows you to incorporate arbitrarily complex geometries.
The supported CAD formats are:

- **STL** - Stereolithography binary format
- **PLY** - Polygon ASCII format

STL and PLY files describe a geometry as a tessellation, providing a set of vertices and faces of triangular or quadrangular surfaces to approximate the volume. While some STL and PLY files also contain additional information such as material and color, TOPAS does not currently accept such information. The STL and PLY files you provide to TOPAS must contain only the tessellation information. Internally, TOPAS represents this component as a `G4TessellatedSolid`.

Most CAD systems allow direct export of parts to the above formats. If your CAD system does not support one of those formats, you may be able to convert from some other CAD format by using a free conversion tool such as MeshLab.

```plaintext
# Common parameters: type of geometry, position, and rotation
s:Ge/MyPartFromCAD/Type = "TsCAD"
s:Ge/MyPartFromCAD/Parent = "World"
s:Ge/MyPartFromCAD/Material = "G4_WATER"
d:Ge/MyPartFromCAD/TransX = 0.0 cm
d:Ge/MyPartFromCAD/TransY = 0.0 cm
d:Ge/MyPartFromCAD/TransZ = 0.0 cm
d:Ge/MyPartFromCAD/RotX = 0.0 deg
d:Ge/MyPartFromCAD/RotY = 0.0 deg
```
TOPAS does not automatically know where the center of your CAD component will be. This is affected by how your CAD system manages coordinates. For example, some CAD software exports the STL by relocating the volume to the first positive octant of its coordinate system. You may have to adjust the TransX/Y/Z parameters of your component to center it as desired.

Above, A plastic scintillator with customized groove. Left: CAD, Right: TOPAS

**Aperture**

An aperture is a component used to shape the lateral penumbra of a (generally) double-scattered proton beam. It is basically a block of brass with a hole cut out from the middle, in the shape of the treatment volume. The purpose is to block the beam outside the desired irradiation path. TOPAS models the aperture by connecting the aperture file points to create a polygon and then extruding this polygon in Z to cut out the aperture hole.
A typical implementation of an apertures in TOPAS is given below followed by a more detailed description of each option (see `ScatteringNozzle.txt` example):

```plaintext
d:Ge/Aperture/Type = "TsAperture"
d:Ge/Aperture/Parent = "Snout"
d:Ge/Aperture/Material = "Brass"
d:Ge/Aperture/RMax = 4.5 cm
d:Ge/Aperture/HL = 2.5 cm
d:Ge/Aperture/TransX = 0.0 cm
d:Ge/Aperture/TransY = 0.0 cm
d:Ge/Aperture/TransZ = -13.0 cm
d:Ge/Aperture/RotX = 0.0 deg
d:Ge/Aperture/RotY = 0.0 deg
d:Ge/Aperture/RotZ = 0.0 deg
d:Ge/Aperture/InputFile = "ApertureFileIn.ap" # Match exact case
d:Ge/Aperture/FileFormat = "XYCoordinates" # XYCoordinates or MGH
d:Ge/Aperture/PrintPoints = "True" # Print points to console
```

**FileFormat** has two options:

- "**XYCoordinates**" takes is a simple list of points. The first line defines how many points there are in the file, each following line in the file is a comma separated x,y pair, such as:
  - numberOfPoints
  - x1,y1
  - x2,y2
  - ...
  - xN,yN

  N = numberOfPoints is the number of data points (xi, yi). This is a required condition. The units of the coordinates are millimeter.

- "**MGH**" takes the milling data produced by the MGH machine shop. It consists of the same information as the "**XYCoordinates**" option, but with more overhead, such as:
  - patientName
- Warning message about not fabricating this file
- someDoubleValue
- someIntValue
- M (this is a number of dummy points, this amount of points will be skipped)
- x1 y1 x2 y2 ... xM yM
- N
- x1 y1 x2 y2 ... xN yN

Data pairs are listed in a simple space-separated list. The units are in centimeter.

**Compensator**

A compensator is a component that is used to shape the distal edge of a proton beam by placing a varying amount of material in the beam path, usually behind the aperture. An example compensator is shown below in top and side views. The compensator consists of a material that is to be placed in the beam to attenuate the beam (usually lexan) and a number of drill holes that were drilled into the compensator.

A typical compensator has the following parameters (see *ScatteringNozzle.txt* example):

```plaintext
<s:Ge/Compensator/Type = "TsCompensator"
<s:Ge/Compensator/Parent = "Snout"
<s:Ge/Compensator/Material = "CompensatorLucite"
<d:Ge/Compensator/RMax = 5.5. cm
<d:Ge/Compensator/TransX = 0. cm
<d:Ge/Compensator/TransY = 0. cm
/dc:Ge/Compensator/Thickness = 0. cm # will be reset to actual thickness when compensator is read in. This allows other parameter files to access this variable thickness.
<d:Ge/Compensator/InvHL = -0.5 * Ge/Compensator/Thickness cm
<d:Ge/Compensator/TransZ = -15.5 cm + Ge/Compensator/InvHL # Allows centering regardless of thickness
```

11.9. Specialized Components 79
Thickness has the special parameter type, \( dc \), where the \( c \) means this dimensioned double is changeable, that is, it can change on the fly based on what exact compensator is read in. Other parameters can then take this thickness into account when the perform placements.

FileFormat has two options:

- "RowsAndDepths": all sizes are in millimeters:
  - numberOfRows
  - MainCylinderThickness
  - DrillHoleDiameter
  - \( n_1 \) deltaX1 X1 Y1
  - D1 D2 ... Dn1
  - \( n_2 \) deltaX2 X2 Y2
  - D1 D2 ... Dn2
  - ...
  - \( n_N \) deltaXn Xn Yn
  - D1 D2 ... DnN

NumberOfRows = \( N \) defines how many rows of drill holes there are (in Y), the MainCylinderThickness. The DrillHoleDiameter is the diameter of the drill hole, we approximate this by a hexagon. The values \( n_i \) are the number of drill holes in X for each row of drill holes in Y, deltaXi defines the step size (and direction) and Xi and Yi are the starting position of the drilling for this row.

- "MGH": all sizes are in inches:
  - Some line
  - numberOfRows
  - dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble
  - MainCylinderThickness
  - dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble dummyDouble
  - DrillHoleDiameter
  - \( n_1 \) deltaX1 X1 Y1
  - D1 D2 ... Dn1
  - \( n_2 \) deltaX2 X2 Y2
  - D1 D2 ... Dn2
  - ...
– nN deltaXn Xn Yn
– D1 D2 ... DnN

Method has four options:

• "ExtrudedSolid" builds the compensator from a set of extruded solids. This is the most reliable and efficient technique.

• "Polyhedra" carves hexagon shapes out of the compensator. This method has an extra check to adjust the position of each starting drill hole of each row to compensate for rounding inaccuracies produced by the drilling machine:

```
d:Ge/Compensator/XTolerance = 1. mm

d:Ge/Compensator/YTolerance = 1. mm
```

• "SubtractionCylinders" builds the compensator by subtracting drill hole cylinders from the overall compensator cylinder. This technique gives the most perfect representation of the drilling process, however the added precision is insignificant, while particle navigation time is increased. Note that if you want to visualize this form of compensator, you should use RayTracer, as this is the only Geant4 visualization drivers that can correctly render boolean operations.

• "UnionCylinders" builds the compensator by first creating a union solid of all the holes, and then subtracting this union solid from the overall compensator cylinder. This technique is similar to "SubtractionCylinders" but slightly more efficient. Note that if you want to visualize this form of compensator, you should use RayTracer, as this is the only Geant4 visualization drivers that can correctly render boolean operations.

**Patient Components**

TOPAS currently supports the following Patient Component types:

<table>
<thead>
<tr>
<th>Geometry Component</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient in DICOM Format</td>
<td>TsDicomPatient</td>
</tr>
<tr>
<td>Patient in XiO Format</td>
<td>TsXioPatient</td>
</tr>
</tbody>
</table>

It is also necessary to define how to convert the imaging data to material data, following a *Imaging to Material Conversion* scheme.

**Common Parameters**

Many of the parameters for Patient Components are common to both TsDicomPatient and TsXioPatient. These are described here.

To perform Monte Carlo simulation, TOPAS needs to map each voxel of the patient image to a material, density and, for useful graphics, a color. You specify how to do this by telling TOPAS which *Imaging to Material Conversion* to use.

To dump your file’s raw imaging values to the console:

```
b:Ge/Patient/DumpImagingValues = "True"
```

Set any parent you like, but it is often convenient to place patient into a group component which can then be rotated to represent couch setup:

```
s:Ge/Patient/Parent = "PatientGroup"
```
TOPAS Documentation, Release 3.0

**Warning:** We will eventually get rid of the Ge/Patient/Material parameter, but for now, you need to provide one, though it doesn't really matter what value it has. The actual voxel materials will come from the Imaging to Material Conversion.

Even though a large number of materials are defined in your HU conversion file, TOPAS will only create those materials that are actually used in your CT image. In the 4DCT case, if any image introduces new materials that were not in the first image, Geant4 will be unable to proceed (it cannot load new materials after physics has initialized). TOPAS will exit with a warning message advising you to set the parameter:

```
b:Ge/Patient/PreLoadAllMaterials = "True"
```

Startup will then be slower, since TOPAS will preload the full set of materials defined in your HU conversion file, but your 4DCT will then work.

For single slice thickness images, scoring will use the same voxel divisions as your CT image. For multiple slice thicknesses, scoring will not know what divisions to use unless you explicitly specify these in your scoring parameters, such as:

```
i:Sc/MyScorer/XBins = 512
i:Sc/MyScorer/YBins = 512
i:Sc/MyScorer/ZBins = 256
```

The built-in Geant4 visualization tools do not perform well when a complex voxel structure is loaded. To make visualization more successful, several additional parameters are provided.

There is generally little value in showing all pixels of the image at once. Each slice just covers up the last slice. To instead show only a specific set of slices in any dimension:

```
iV:Gr/Patient/ShowSpecificSlicesZ = 4 1 3 9 12 # will only show slices 1, 3, 9 and 12.
```

Similar slicing is allowed in X and Y. Three special values are also allowed:

```
iV:Gr/Patient/ShowSpecificSlicesZ = 1 0 # means show all slices
iV:Gr/Patient/ShowSpecificSlicesZ = 1 -1 # means only show center slice
iV:Gr/Patient/ShowSpecificSlicesZ = 1 -2 # means only first, center and last slice
```

The following will result in a display that shows 27 pixels comprising the boundaries and center of the image. This allows you to see the overall placement of the image and see the individual voxel size:

```
iV:Gr/Patient/ShowSpecificSlicesX = 1 -2 # means only show center slice
iV:Gr/Patient/ShowSpecificSlicesY = 1 -2 # means only show center slice
iV:Gr/Patient/ShowSpecificSlicesZ = 1 -2 # means only show center slice
```

Another option allows you to specify the maximum number of voxels to show. If the total number of voxels is greater than this limit, TOPAS will just draw the overall DICOM outline:

```
i:Gr/ShowOnlyOutlineIfVoxelCountExceeds = 10000
```

By default, OpenGL graphics switches its fast “Stored” mode to its more memory efficient “Immediate” mode when the graphics scene gets very complicated. When this switch occurs, the current version of Geant4 has a bug such that part of the image is lost. To prevent this from impacting DICOM images, you can set a threshold at which Geant4 will use Immediate mode from the start:

```
i:Gr/SwitchOGLtoOGLIifVoxelCountExceeds = 10000 # Above this limit, switch OpenGL Graphics to Immediate mode
```
Patient in DICOM Format

DICOM import is handled through the GDCM package, which is pre-built into TOPAS.

See the DoseToCT.txt and DoseTo4DCT.txt examples of how to use TsDicomPatient. Note that before running this example, you must unzip the included DICOM files.

You specify the name of a directory containing one or more dcm files (one for each slice):

```plaintext
g:Ge/Patient/DicomDirectory = "DICOM_Box"
```

To specify 4DCT, you can have DicomDirectory change under control of a Time Feature.

Files of other types in this directory will be ignored. Exact titles of the dcm files are not important as TOPAS will re-order them based on the slice ordering information inside the DICOM headers.

By default, Topas will only consider dcm files that are from CT. This can be adjusted by:

```plaintext
sv:Ge/Patient/DicomModalityTags = 1 "CT" # defaults to just CT
```

Other modality tags are, for example, "MR" for Magnetic Resonance and "US" for Ultrasound. A complete list can be found here.

Patient positioning information from the DICOM file is not currently used. You must position as you would for any TOPAS component:

```plaintext
d:Ge/Patient/TransX=0. m
d:Ge/Patient/TransY=0. m
d:Ge/Patient/TransZ=0. m
d:Ge/Patient/RotX=0. deg
d:Ge/Patient/RotY=0. deg
d:Ge/Patient/RotZ=0. deg
```

TOPAS can read DICOM RT Structure Sets. A structure set is an extra file in the DICOM directory that provides information on structures such as organs, tumors, PTVs, etc. that have been outlined (contoured) in the planning process. The data is stored as a set of polygons, up to one per slice for each contoured structure. TOPAS can color code DICOM components according to this structure information and can filter scoring based on these structures (see the filter: OnlyIncludeIfInRTStructure).

To make TOPAS color the voxels by structure:

```plaintext
sv:Ge/Patient/ColorByRTStructNames = 2 "R_LUNG" "L_LUNG"
sv:Ge/Patient/ColorByRTStructColors = 2 "yellow" "red"
```

- If the structure name includes a space, substitute an underscore in the parameter. So, for example, if the structure name is “R LUNG”, you should supply the parameter as “R_LUNG”.
- If you don’t actually know what structures are included in your DICOM, just try providing in ColorByRTStructNames. TOPAS will give you an error message that includes a list of the known structure names.
- To allow easy testing of this feature in simple DICOM examples that don’t really have any structures, the following parameter will “fake” an RT structure set, assigning the given structure to all voxels in the lower XY quadrant:

```plaintext
b:Ge/Patient/FakeStructures = "True"
```
Patient in XiO Format

The XiO patient is a specific implementation of a patient geometry. It requires a binary file containing a list of Hounsfield units for each voxel of a patient in “short little endian” format.

See the DoseToCT.txt example of how to use TsXiOPatient.

Specify file directory and file name:

```plaintext
s:Ge/Patient/InputDirectory = "./"
s:Ge/Patient/InputFile = "ctvolume.dat" # match exact case
```

To specify 4DCT, you can have InputDirectory or InputFile change under control of a Time Feature.

You must position as you would for any TOPAS component:

```plaintext
d:Ge/Patient/RotX = 0. deg
d:Ge/Patient/RotY = 90. deg
d:Ge/Patient/RotZ = 0. deg
d:Ge/Patient/TransX = 1.5 mm
d:Ge/Patient/TransY = 3.3 mm
d:Ge/Patient/TransZ = 4.2 mm
```

XiO Format does not contain voxel number and size information. You must specify it as follows.

Number and size of voxels in X and Y:

```plaintext
i:Ge/Patient/NumberOfVoxelsX = 25
i:Ge/Patient/NumberOfVoxelsY = 25
d:Ge/Patient/VoxelSizeX = 2.0 mm
d:Ge/Patient/VoxelSizeY = 2.0 mm
```

Number and size of Voxels in Z are vectors rather than single values since XiO file may have multiple slice thickness sections.

If there is only one slice thickness in your image, just specify one element:

```plaintext
iv:Ge/Patient/NumberOfVoxelsZ = 1 10
dv:Ge/Patient/VoxelSizeZ = 1 2.5 mm
```

If there are multiple slice thicknesses in your image, specify number and thickness of voxels in each section. For example, a 30 slice image that has 10 slices of 2.5 mm and then 20 slices of 1.25 mm:

```plaintext
iv:Ge/Patient/NumberOfVoxelsZ = 2 10 20
dv:Ge/Patient/VoxelSizeZ = 2 2.5 1.25 mm
```

Imaging to Material Conversion

You are free to write your own converter, including approaches that use alternative imaging modalities (e.g. MRI, pCT, ultrasound), or that use more than one image (e.g. Dual Energy CT, Multi-Energy CT). To write your own converter, see Custom Imaging to Material Conversion.

TOPAS provides one built-in converter, that follows the most common method used in proton therapy (PubMed):


This converter follows the technique developed by Schneider to assign materials based on a single CT image file containing Hounsfield Unit (HU) values. It is selected using:
The HU conversion parameters are typically stored in a separate parameter file:

**includeFile** = HUtoMaterialSchneider.txt

An example of such a HU conversion parameter file is examples/DICOM/HUtoMaterialSchneider.txt.

The first set of parameters in the HU file are used to calculate density:

- **DensityCorrection**:
  - One value for every possible HU value.
  - Values start from **Ge/Patient/MinImagingValue** which defaults to -1000

- **SchneiderHounsfieldUnitSections**:
  - Specifies how to break up the entire set of HU units into several density calculation sections. The HU conversion formula then uses different correction factors for each of these sections.
  - The total range (last value minus first value) must equal the number of values in **DensityCorrection**.
  - In the above example, the 8 values define 7 sections:
    - Section 1: -1000 to -99
    - Section 2: -98 to 14
    - ...
    - Section 7: 2995 to 2996

- **SchneiderDensityOffset**, **SchneiderDensityFactor** and **SchneiderDensityFactorOffset**:
  - Must have one value for each of the density calculation sections, so length must be one less than the length of **SchneiderHounsfieldUnitSections**

Thus, for any specific HU number, we can extract the appropriate:

- **DensityCorrection**
- **SchneiderDensityOffset**
- **SchneiderDensityFactor**
- **SchneiderDensityFactorOffset**

And use these in the Schneider formula:

- Density = ( Offset + ( Factor * ( FactorOffset + HU[-1000,2995] ) ) ) * DensityCorrection

The second set of parameters in the HU file are used to calculate material name and graphics color:

- **SchneiderHUtoMaterialSections**
- **SchneiderElements**
iv: SchneiderHUToMaterialSections:

- Specifies how to break up the entire set of HU units into several material name assignment sections.
- The total range (last value minus first value) must equal the number of values in DensityCorrection.
- In the above example, the 26 values define 7 material name assignment sections:
  - Section 1: -1000 to -949
  - Section 2: -50 to -119
  - ...
  - Section 26: 2995 to 2996

sv: SchneiderElements:

- Specifies all of the elements that will be used in the patient.
- All patient materials must be composed from combinations of this set of elements.

uv: SchneiderMaterialsWeight1 through SchneiderMaterialsWeight26:

- There should be one of these parameters for each of the material name assignment sections. The length of SchneiderMaterialsWeight must equal the length of SchneiderElements.
- Each value in SchneiderMaterialsWeight tells what proportion of the given element in SchneiderElements to use in this material.
- In our SchneiderMaterialsWeight2 parameter, the values: 0.103 0.105 0.031 0.749 mean:
  - 10.3 percent of the first element, Hydrogen
  - 10.5 percent of the second element, Carbon
  - 3.1 percent of the second element, Nitrogen
  - 74.9 percent of the second element, Oxygen

dv: SchneiderMaterialMeanExcitationEnergy:

- You may optionally provide this parameter to override the default mean excitation energies of some or all of the materials.
- There should be one value for each material name assignment section.
- To use the default mean excitation energy for a particular material, enter that value as 0.
- For example, the following just overrides defaults for two out of 26 assignment sections:

```
iv: Ge/Patient/SchneiderMaterialMeanExcitationEnergy = 26 88.8 0. 77.7 . 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. eV
```

iv: Gr/Color/PatientTissue1:

- Specifies what colors should be assigned to each of the materials.
- There should be one of these parameters for each of the SchneiderHUToMaterialSections.
• The three values specify the Red, Green and Blue components of the color.

Putting it all together, we have now specified density, material name, color and, optionally, mean excitation energy, for each of the Hounsfield numbers in the patient.

You can review the materials definitions that TOPAS created based on your patient file and the HU conversion settings. The following parameter tells TOPAS to dump parameters to a file:

```
Ts/DumpNonDefaultParameters = "True"
```

For each HU number that was used in the patient file, you will see a set of parameters starting with `Ma/PatientTissueFromHU` followed by an HU number. For example, for HU number -295, you may see:

```
Ma/PatientTissueFromHU-295/Component = 9 Hydrogen Carbon Nitrogen Oxygen Phosphorus Sulfur Chlorine Sodium Potassium
Ma/PatientTissueFromHU-295/Fractions = 9 0.103 0.105 0.031 0.749 0.002 0.003 0.003 0.002 0.002
Ma/PatientTissueFromHU-295/Density = 0.707487 g/cm³
Ma/PatientTissueFromHU-295/DefaultColor = PatientTissue2
```

where you then follow the `DefaultColor` parameter named `PatientTissue2` to see that `Gr/Color/PatientTissue2` is `3 100 0 0` which means a mixture of 100 percent Red, 0 percent green, 0 percent blue.
CHAPTER 12

Particle Sources

Introduction

We allow any number of particle sources (zero, one or many) with no limitation on how they can be mixed.

We provide four different types of particle sources, each with many options:

• Beam Parameterization
• Emittance Parameterization
• Isotropic
• Phase Space

And you may also write your own entirely new particle source (see Custom Particle Sources).

The position of the source is always the center of an associated Geometry Component. This is in keeping with the general TOPAS paradigm that all geometrical information resides in Geometry Components. We know that this may feel odd to experienced Geant4 users who are used to setting beam directions irrespective of any geometry volumes, but the TOPAS paradigm enables sources, components, scorers and even fields to all move together in an internally consistent manner.

The Geometry Component associated with a Particle Source is often a Group Component. Such components have a center position and orientation but no actual shape or extent. The Particle Source is placed at this center position and orientation. If the Component is some other Type, such as a TsBox or TsCylinder, the Particle Source still only takes center position and orientation from this Component. None of the other aspects of the Component, such as the Component’s shape or size, have any impact on the Particle Source. So, for example, the shape and size of a Beam source is set by various BeamPosition parameters, not by the Component’s shape or size.

Some examples place the source at a vacuum window at the entrance to a nozzle. The source then moves as the nozzle moves.

Particle Names

Throughout TOPAS, particle names can take the following forms (case does not matter):
A simple string such as

- "proton"

A string describing an ion with arguments Z, A, and optionally Charge, such as:

- "GenericIon(6,12,6)"
- "GenericIon(6,12)" - Charge defaults to Z, that is, the ion is fully stripped
- When used to filter sources, ions must be fully stripped (this is the only kind of ion that Geant4’s primary particle generation supports).
- When used to filter scoring, ions can have any Charge, and any of the arguments can have wildcard value *
  so, for example, “GenericIon(6,*,*)” will score any Carbon ion (any A and any Charge).

An integer PDG ID code, though still contained in a string parameter, such as

- “11”
- PDG ID codes are as defined by the Particle Data Group

When PDG code has 10 digits and starts with 100, this is passed to Geant4 either as the appropriate Geant4 light ion name (“alpha”, “deuteron”, “He3” or “triton”) or as GenericIon(Z,A) where:

- Characters 4-6 give Z
- Characters 7-9 give A
- Character 10 gives Isomer level (not used)

The full set of known particles depends on the physics you have defined. Here are some common values, with associated PDG codes:

<table>
<thead>
<tr>
<th>Particle</th>
<th>PDG code</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;proton&quot;</td>
<td>&quot;2212&quot;</td>
</tr>
<tr>
<td>&quot;neutron&quot;</td>
<td>&quot;2112&quot;</td>
</tr>
<tr>
<td>&quot;e-&quot;</td>
<td>&quot;11&quot;</td>
</tr>
<tr>
<td>&quot;e+&quot;</td>
<td>&quot;-11&quot;</td>
</tr>
<tr>
<td>&quot;gamma&quot;</td>
<td>&quot;22&quot;</td>
</tr>
<tr>
<td>&quot;He3&quot;</td>
<td>&quot;100002003&quot;</td>
</tr>
<tr>
<td>&quot;alpha&quot;</td>
<td>&quot;100002004&quot;</td>
</tr>
<tr>
<td>&quot;deuteron&quot;</td>
<td>&quot;100001002&quot;</td>
</tr>
<tr>
<td>&quot;triton&quot;</td>
<td>&quot;100001003&quot;</td>
</tr>
<tr>
<td>&quot;opticalphoton&quot;</td>
<td>&quot;0&quot; (PDG group has no code for this particle)</td>
</tr>
<tr>
<td>&quot;geantino&quot;</td>
<td>&quot;0&quot; (sees transportation processes but no physics, no PDG code)</td>
</tr>
<tr>
<td>&quot;chargedgeantino&quot;</td>
<td>&quot;0&quot; (same as above but with charge, no PDG code)</td>
</tr>
</tbody>
</table>

**Beam Sources**

By default there is a single source named Demo centered on a Component named BeamPosition that is placed at one end of the World. The beam shape is an Ellipse. Each of these parameters is described in detail below:

```plaintext
a:S0/Demo/Type = "Beam" # Beam, Isotropic, Emittance or PhaseSpace
d:S0/Demo/BeamParticle = "proton"
d:S0/Demo/BeamEnergy = 169.23 MeV
u:S0/Demo/BeamEnergySpread = 0.757504
a:S0/Demo/BeamPositionDistribution = "Gaussian" # None, Flat or Gaussian
a:S0/Demo/BeamPositionCutoffShape = "Ellipse" # Rectangle or Ellipse (if Flat or Gaussian)
```
Where the default definition of BeamPosition is:

```
so/BeamPosition/Parent="World"
so/BeamPosition/Type="Group"
so/BeamPosition/TransX=0. m
so/BeamPosition/TransY=0. m
so/BeamPosition/TransZ=so/World/HLZ m
so/BeamPosition/RotX=180. deg
so/BeamPosition/RotY=0. deg
so/BeamPosition/RotZ=0. deg
```

Details on BeamEnergySpread:

- The number is unitless because we find it more convenient generally to speak of the spread in terms of percentage of the mean energy, rather than as an absolute number. We could have chosen either representation, but this one seemed most consistent with what we see from other beam modeling applications.

- This is a standard deviation. So the code we have is:

  ```c
  fEnergySpread = BeamEnergySpread * fEnergy / 100.;
  p.kEnergy = CLHEP::RandGauss::shoot(fEnergy, fEnergySpread);
  ```

- So, for example, if you want a spread of 0.2 MeV, and your energy is 153 MeV, set BeamEnergySpread to:

  \[
  0.2 \text{ MeV} / 153 \text{ MeV} \times 100 = 0.13
  \]

To run generate histories using this demo source, set its number of histories to some value:

```
so/Demo/NumberOfHistoriesInRun = 10
```

We recommend that you not use So/Demo for any serious work. This demonstration source is just there for simple demonstrations. For any serious work, please define your own source so that you do not just accidentally inherit any of the characteristics of our Demo source. Source characteristics vary greatly from one application to another. There is no meaningful “default” value that we can set for you.

So when you set out on your own work, define a new source name, such as:

```
s:mySource/BeamParticle = "proton"
s:mySource/BeamEnergy = 200. MeV
i:mySource/NumberOfHistoriesInRun = 100
...```

You can provide an energy spectrum instead of a fixed energy by setting the following to "Discrete" or "Continuous":

---

**12.2. Beam Sources**
and providing energies and weights as:

```plaintext
dv:So/MySource/BeamEnergySpectrumValues = 3.5 100.1 150. MeV
du:So/MySource/BeamEnergySpectrumWeights = 3.20 .60 .20
```

An example is in `Spectrum.txt`.

Any source that has `NumberOfHIsoriesInRun` greater than zero will contribute primary particles.

The beam is emitted along the Z axis of the beam's `Component` and may have some spread along the X and Y axes.

For `Type = "Beam"`, the beam shape can be further described by a set of parameters that control the position distribution of the start of the beam:

```plaintext
d:So/Demo/BeamPositionDistribution = "Gaussian" # None, Flat or Gaussian
d:So/Demo/BeamPositionCutoffShape = "Ellipse" # Rectangle or Ellipse (if Flat or Gaussian)
d:So/Demo/BeamPositionCutoffX = 10. cm # X extent of position (if Flat or Gaussian)
d:So/Demo/BeamPositionCutoffY = 10. cm # Y extent of position (if Flat or Gaussian)
d:So/Demo/BeamPositionSpreadX = 0.65 cm # distribution (if Gaussian)
d:So/Demo/BeamPositionSpreadY = 0.65 cm # distribution (if Gaussian)
```

and a set of parameters that control how the beam spreads out from that start position:

```plaintext
d:So/Demo/BeamAngularDistribution = "Gaussian" # None, Flat or Gaussian
d:So/Demo/BeamAngularCutoffX = 90. deg # X cutoff of angular distrib (if Flat or Gaussian)
d:So/Demo/BeamAngularCutoffY = 90. deg # Y cutoff of angular distrib (if Flat or Gaussian)
d:So/Demo/BeamAngularSpreadX = 0.0032 rad # X angular distribution (if Gaussian)
d:So/Demo/BeamAngularSpreadY = 0.0032 rad # Y angular distribution (if Gaussian)
```

The `Cutoff` and `Spread` parameters are applied symmetrically.

You will note that for Gaussian beams, the position and angular distribution are controlled both by `Spread` and by `Cutoff` parameters. The `Spread` control the standard deviation of the Gaussian, while the `Cutoff` cut off the tails (which would otherwise be infinite). Inside TOPAS, when the Gaussian formula generates a starting point outside of this cutoff, that starting point is rejected and instead the random function is thrown again until a value is found that is within the specified cutoff.

## Emittance Sources

Emittance sources provide 4 ways of sampling particles’ position (X and Y) and momentum direction (X’ and Y’) on a plane. Specify source type as:

```plaintext
d:So/MySource/Type = "emittance"
```

To sample particle position and momentum direction from “Gaussian” (bivariate 2D gaussian) per axis (see the `Emittance_Gaussian.txt` example):

```plaintext
d:So/MySource/Distribution = "BiGaussian" # distribution name
d:So/MySource/SigmaX = 0.2 mm # std of x positions
d:So/MySource/SigmaXprime = 0.032 # std of x’, note that it’s unitless. 1 equals to 1.140 rad.
```
Alternative distributions are TWISS Gaussian, TWISS Kapchinskij-Vladimirskij (KV), and TWISS Waterbag (based on the Courant-Snyder invariant ellipse in the following figure for X axis only):

![TWISS ellipse equation](image)

Additional parameters specify the TWISS alpha, beta and gamma as shown in the following from `Emittance_Twiss.txt`:

```
s:So/MySource/Distribution = "twiss_gaussian" # "twiss_gaussian", "twiss_kv" or "twiss_waterbag"
s:So/MySource/AlphaX = 0.2
d:So/MySource/BetaX = 600.0 mm
d:So/MySource/EmittanceX = 0.01 mm # we don’t multiply pi intrinsically.
s:So/MySource/AlphaY = 2.5
d:So/MySource/BetaY = 1400.0 mm
d:So/MySource/EmittanceY = 0.02 mm
# below for twiss gaussian option only
#0.9 means that 90 % particles will be included in ellipse circle
u:So/MySource/ParticleFractionX = 0.90
u:So/MySource/ParticleFractionY = 0.90
```
The energies and species of the emitted particles can be specified using the same parameters available to the Beam Sources.

The following images based on the Emittance_Twiss.txt example show how particle position and momentum directions are sampled and the shape of the beam spots (purple). The blue solid lines in X (red dots) and Y (green dots) represent the Courant-Snyder invariant ellipse. For example, 90% of particles are sampled from the given emittance ellipse in (a).

**Isotropic Sources**

Isotropic sources emit particles uniformly from the center of the specified Component.

Specify source type as:
The energies and species of the emitted particles can be specified using the same parameters available to the Beam Sources.

**Phase Space Sources**

Phase Space refers to the technique of saving or replaying a set of particles crossing a given surface.

- When one saves a phase space, one defines a surface and then saves the position, particle type, energy and momentum of some or all particles crossing that surface.
- When one replays a phase space, one starts a set of particles from the saved positions, with the saved particle types, energy and momentum.

Phase Space enables separating two parts of a simulation or analysis job, and can be used to transfer sets of particles among different codes.

Each phase space must come as two related files (with same file name but different file extensions):

- A `.header` file tells the number of histories, the number of saved particles and the order of information in the `.phsp` file
- A `.phsp` file contains all the details of all the saved particles

We support three formats for Phase Space (and TOPAS automatically figures out the format of your `.phsp` file by studying the related `.header` file):

- ASCII provides particle information in a human-readable text file, which data encoded as a series of columns of text. The header file tells the contents and column order per particle.
- Binary provides the same information as ASCII, but in a much more compact format, with data encoded in a stream of bytes. The header file tells the contents and byte order per particle. Use Binary in cases where the ASCII format produces excessively large files.
- Limited is an alternate binary format compatible with some legacy codes. It has fewer options for what data can be expressed, but is compatible with codes such as that used by Varian for their TrueBeam phase space files. Use Limited format only when you need to exchange phase space with legacy codes.

Note that while our Phase Space Scorer lets you also write phase space to ROOT files, we do not provide the capability read phase space back in from this format. For more details, see Phase Space Format.

Phase Space sources ignore the parameters starting with “Beam” and instead use:

- `:So/MySource/Type = "PhaseSpace"
- `:So/MySource/Component = "World"` # coordinate system of phase space. Usually "World"
- `:So/MySource/PhaseSpaceFileName = "ASCIIOutput"` # match exact case

TOPAS will look for header and phsp files with the given `PhaseSpaceFileName`.

You can generate some sample data by running any of the examples: WriteASCII.txt, WriteBinary.txt or WriteLimited.txt.

When using phase space sources, it is important to decide how you want to handle a special case we call “Empty Histories.” Recall that when a phase space is first recorded, for a given Original History, the set of resulting particles that cross the phase space surface:

- may include the primary particles, or
- may include a mix of primary and secondary particles, or
• may include only secondary particles, or
• may include no particles at all. We refer to this last case as an “Empty History.”

When you subsequently use this file as a Phase Space Source, you need to decide how you want TOPAS to handle Empty Histories. If you’re just calculating sums, it doesn’t matter. The Empty Histories contribute nothing to the sum anyway. But if you’re calculating statistical quantities, such as Mean, then these Empty Histories matter. Imagine you want to know the mean dose per Original History. If half of the Original Histories never made it to the phase space file, the decision of whether or not to include these Empty Histories will give a factor of two difference in the calculated Mean Dose per History.

Depending on your use case you may or may not want to include these Empty Histories. It comes down to whether the statistics you want to calculate are:

• per Original History, or
• per Original Histories that Reached Phase Space

You control this with:

```
S:So/MySource/PhaseSpaceIncludeEmptyHistories = "False" # defaults to false
```

TOPAS ASCII and Binary phase space format headers show all of the relevant information:

• Number of Original Histories
• Number of Original Histories that Reached Phase Space
• Number of Scored Particles

Limited phase space format header does not give:

• Number of Original Histories that Reached Phase Space
• so the only way to get that in Limited format is to first read through the entire phsp file and count how many histories contributed there.

TOPAS provides an option to check that the values in the header match what is in the file:

```
S:So/MySource/PhaseSpacePreCheck = "True" # defaults to true
```

For TOPAS ASCII and Binary formats, this is a thorough safety check. It will catch any cases where the files have somehow become corrupted (as could happen, for example, if you are doing a very long phase space writing job and the output disk becomes full during some part of the job). For Limited format, the check is still helpful but less thorough as the header file provides incomplete information. In Limited format, if you want to include Empty Histories, the check is required as it is the only way TOPAS can figure out how many Empty Histories there were.

If the phase space you are replaying came from a TOPAS job, the particle starting positions in that file will have been defined relative to the World Component. Set the Component parameter above to "World". If you want to offset these particles to some other center or orientation, choose a Component that has the new desired center and orientation (reuse some existing Component, or define a new Group Component just for this purpose). If the phase space you are replaying did not come from TOPAS, there is no automatic way to know what coordinate system was used. It will be up to you to choose a Component that has this appropriate coordinate system.

You can optionally tell the phase space source to ignore parts of its position information:

```
b:So/MySource/PhaseSpaceIgnoreXPos = "True" # start all particles at Component’s X-axis
b:So/MySource/PhaseSpaceIgnoreYPos = "True" # start all particles at Component’s Y-axis
b:So/MySource/PhaseSpaceIgnoreZPos = "True" # start all particles at Component’s Z-axis
```
That coordinate of the particle position then just exactly matches the Component center.

You can optionally invert any of the phase space axes by:

```
b:So/MySource/PhaseSpaceInvertXAxis = "True"
b:So/MySource/PhaseSpaceInvertYAxis = "True"
b:So/MySource/PhaseSpaceInvertZAxis = "True"
```

In most cases you will instead want to just rotate the source component. However if the handedness of your source phase space is incorrect, one of these invert options will be necessary.

By default, a PhaseSpace source will run all of the histories in the file. To run all of the histories multiple times:

```
i:So/MySource/PhaseSpaceMultipleUse = 2 # reuse this phase space multiple times
```

If you set PhaseSpaceMultipleUse to zero, the number of histories in the file will be ignored, and we will instead run the exact number from:

```
i:So/MySource/NumberOfHistoriesInRun
```

This may mean only partial use of the phase space file, or partial reuse to get the right number of histories.

- If your data was generated with time dependence, partial reuse of phase space may not give valid results (you may be playing back only a part of the time sequence). Many more details on controlling number of histories are found in *Time mode*.

- Partial reuse of phase space can not include Empty Histories. There is no statistically valid way to handle these empty histories when the phase space file is only partially used (since one does not know where in the phase space order these Empty Histories would have occurred).

For efficiency, the phase space file will be read in chunks of 10,000 particles at a time. Advanced users may find some reason to adjust this buffer size (though I can’t think of any):

```
i:So/MySource/PhaseSpaceBufferSize = 1000000
```

Take care when mixing Phase Space Sources with *Time Features*. While TOPAS can save the current TOPAS time to a phase space file, this time is not automatically applied when reading particles back in from phase space. Thus, if you want to correctly replay source particles that were recorded with time features, it is your responsibility to apply the identical time features during the play back simulation. Some additional notes:

- Do not attempt to change the name of the phase space file over time. Save and replay all particles from a single phase space file.

- Do not use *Random Time Mode*. The randomly generated times during playback will not necessarily match the randomly generated times that were saved to the phase space. Only use *Fixed Time Mode* or *Sequential Time Mode*.

- If your intention is to play back with exactly the same sequence as you had when you generated the phase space file, make sure to set:

```
s:So/MySource/PhaseSpaceIncludeEmptyHistories = "True"
```

otherwise empty histories will put the playback job out of synch with the original job.

A future version of TOPAS will provide more tools to synchronize and check playback time features.

---

12.5. Phase Space Sources
Miscellaneous

Additional Control of Number of Histories

Because TOPAS supports both sequential and random time, there are additional parameters that can control the number of histories in random mode. Read *Time mode* before using these parameters:

```
i:So/Demo/NumberOfHistoriesInRandomJob = 100
d:So/Demo/ProbabilityOfUsingAGivenRandomTime = 1.
```

Filtering Sources

Optionally filter what comes from the source. This is mainly intended for use with saved PhaseSpace, but is applied uniformly to all sources. Syntax is identical to that used for *filtering in Scorers*.

You may write your own additional filters (see *Custom Filters*).

Filter by Charge. Accepts one or more of "Positive", "Negative" or "Neutral":

```
sv:So/MySource/OnlyIncludeParticlesCharged = 1 "Negative"
sv:So/MySource/OnlyIncludeParticlesNotCharged = 1 "Negative"
```

Filter by Atomic Mass or Number:

```
i:So/MySource/OnlyIncludeParticlesOfAtomicMass = 10 # allow all ions of atomic mass 10
i:So/MySource/OnlyIncludeParticlesNotOfAtomicMass = 10
i:So/MySource/OnlyIncludeParticlesOfAtomicNumber = 6 # allow all ions of Carbon
i:So/MySource/OnlyIncludeParticlesNotOfAtomicNumber = 6
```

Filter by Particle’s Initial Kinetic Energy:

```
d:So/MySource/OnlyIncludeParticlesWithInitialKEBelow = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialKENotBelow = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialKE = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialKENot = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialKEAbove = 10. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialKENotAbove = 10. MeV
```

Filter by Particle’s Initial Momentum:

```
d:So/MySource/OnlyIncludeParticlesWithInitialMomentumBelow = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialMomentumNotBelow = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialMomentum = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialMomentumNot = 1. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialMomentumAbove = 10. MeV
d:So/MySource/OnlyIncludeParticlesWithInitialMomentumNotAbove = 10. MeV
```

Filter by Particle Name:

```
sv:So/MySource/OnlyIncludeParticlesNamed = 2 "proton" "neutron"
sv:So/MySource/OnlyIncludeParticlesNotNamed = 2 "proton" "neutron"
```

Particle names are as described *here*.

You may specify more than one filter. For example, to emit protons with initial KE over 100 MeV:
You can invert the results of all previous filters. The following would emit only particles that are Not protons with initial KE over 100 MeV:

```
sv:Sc/MySource/OnlyIncludeParticlesNamed = 2 "proton" "neutron"
d:Sc/MySource/OnlyIncludeParticlesWithInitialKEAbove = 100. MeV # minimum energy
b:Sc/MySource/InvertFilter = "True"
```

Any filter property can be set by *Time Features* if you wish, to produce time-dependent filtering.
CHAPTER 13

Physics

Introduction

In Geant4, physics options are set in pieces of code called “Physics Lists”. A physics list specifies what particles and physics processes are defined, plus various cuts and options. By default, we set TOPAS physics to a list that has been shown to work well for proton therapy research at the Massachusetts General Hospital. This list includes models that handle not only protons but also all secondary particles (neutrons, helium ions, deuterons, tritons, photons, electrons, etc.). The default gives results that closely match a previous custom list that was described in:


but which can no longer be used since that list corresponded to a much earlier Geant4 release.

Advanced users can set their own parameters to override some of these default settings, or can specify entirely different physics lists.

You can choose from two general types of physics lists:

- **Reference Physics Lists** are pre-made, complete lists provided by Geant4.
- **Modular Physics Lists** are lists where you mix and match a set of modules to create a customized complete list.

You can also provide your own physics list using **Custom Physics Lists** (not recommended unless you have significant Geant4 expertise).

You can get a list of what processes are in your currently selected physics list by:

```
$ Ph/ListProcesses = "True"
```

Modular Physics Lists

The default list we provide is a Modular physics list. It is specified by the parameters described [here](#).

The Geant4 EM physics group recommends against setting **EMRangeMin** too low:
• Set to 100. eV or greater when using standard Geant4 EM physics
• Set to 10. eV or greater when using Geant4-DNA physics

If you want to run with no physics, but only the transportation process (useful for some demos and tests), specify the modules in the following special way:

```sv:Ph/Default/Modules = 1 "Transportation_Only"
```

Below is a List of Available Modules with the corresponding Geant4 class names.

The remaining options for the "Geant4_Modular" physics type are:

```d:Ph/Default/CutForAllParticles = 0.05 mm # single range cut to use for all particles
d:Ph/Default/CutForGamma = 0.05 mm # overrides CutForAllParticles for Gamma
d:Ph/Default/CutForElectron = 0.05 mm # overrides CutForAllParticles for Electron
d:Ph/Default/CutForPositron = 0.05 mm # overrides CutForAllParticles for Positron
d:Ph/Default/CutForProton = 0.05 mm # overrides CutForAllParticles for Proton
d:Ph/Default/CutForAlpha = 0.05 mm # overrides CutForAllParticles for Alpha
d:Ph/Default/CutForDeuteron = 0.05 mm # overrides CutForAllParticles for Deuteron
d:Ph/Default/CutForTriton = 0.05 mm # overrides CutForAllParticles for Triton
d:Ph/Default/EMRangeMin = 100. eV # minimum for EM tables
d:Ph/Default/EMRangeMax = 300. MeV # maximum for EM tables
i:Ph/Default/dEdXBins = 220 # number of bins for dEdX tables
i:Ph/Default/LambdaBins = 220 # number of Lambda bins
b:Ph/Default/Fluorescence = "False" # Set to true to turn on Fluorescence
b:Ph/Default/Auger = "False" # Set to true to turn on Auger
b:Ph/Default/PIXE = "False" # Set to true to turn on PIXE
```

### Physics Regions

By default, cuts affect the entire world, but you can optionally divide the world into several regions and can specify different cuts in each region. First, specify which components belong to a given region:

```s:Ge/MyComponent/AssignToRegionNamed = "MyRegion"
```

• All children of this component will also be assigned to that region, unless the child has its own AssignToRegionNamed parameter.
• There is no requirement that all of the components in a given region be contiguous.

Then assign cuts per region by including the region name in the parameter name as in:

```d:Ph/Default/ForRegion/MyRegion/CutForGamma = 0.05 mm
d:Ph/Default/ForRegion/MyRegion/CutForElectron = 0.05 mm
d:Ph/Default/ForRegion/MyRegion/CutForPositron = 0.05 mm
d:Ph/Default/ForRegion/MyRegion/CutForProton = 0.05 mm
```

Cuts do not affect all processes, but only those listed below:

• Energy thresholds for gamma are used in Bremsstrahlung
• Energy thresholds for electrons are used in ionization and e+e- pair production processes Energy thresholds for positrons are used in e+e- pair production process
• Energy thresholds for gamma and electrons are used optionally in all discrete processes
  – Photoelectric effect
  – Compton
- gamma conversion

- Energy thresholds for protons are used in processes of elastic scattering for hadrons and ions defining the threshold for kinetic energy of nuclear recoil

**List of Available Modules**

<table>
<thead>
<tr>
<th>TOPAS Module Name</th>
<th>Geant4 Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>g4h-chargeexchange</td>
<td>G4ChargeExchangePhysics</td>
</tr>
<tr>
<td>g4decay</td>
<td>G4DecayPhysics</td>
</tr>
<tr>
<td>g4em-dna</td>
<td>G4EmDNAPhysics</td>
</tr>
<tr>
<td>g4em-extra</td>
<td>G4EmExtraPhysics</td>
</tr>
<tr>
<td>g4em-livermore</td>
<td>G4EmLivermorePhysics</td>
</tr>
<tr>
<td>g4em-polarized</td>
<td>G4EmLivermorePolarizedPhysics</td>
</tr>
<tr>
<td>g4em-lowep</td>
<td>G4EmLowEPPhysics</td>
</tr>
<tr>
<td>g4em-penelope</td>
<td>G4EmPenelopePhysics</td>
</tr>
<tr>
<td>g4em-standard_opt0</td>
<td>G4EmStandardPhysics</td>
</tr>
<tr>
<td>g4em-standard_opt1</td>
<td>G4EMStandardPhysics_option1</td>
</tr>
<tr>
<td>g4em-standard_opt2</td>
<td>G4EMStandardPhysics_option2</td>
</tr>
<tr>
<td>g4em-standard_opt3</td>
<td>G4EMStandardPhysics_option3</td>
</tr>
<tr>
<td>g4em-standard_opt4</td>
<td>G4EMStandardPhysics_option4</td>
</tr>
<tr>
<td>g4em-standard_WVI</td>
<td>G4EmStandardPhysicsWVI</td>
</tr>
<tr>
<td>g4h-elastic_D</td>
<td>G4HadronDElasticPhysics</td>
</tr>
<tr>
<td>g4h-elastic</td>
<td>G4HadronElasticPhysics</td>
</tr>
<tr>
<td>g4h-elastic_HP</td>
<td>G4HadronElasticPhysicsHP</td>
</tr>
<tr>
<td>g4h-elastic_LEND</td>
<td>G4HadronElasticPhysicsLEND</td>
</tr>
<tr>
<td>g4h-elastic_XS</td>
<td>G4HadronElasticPhysicsXS</td>
</tr>
<tr>
<td>g4h-elastic_H</td>
<td>G4HadronElasticPhysics</td>
</tr>
<tr>
<td>g4h-inelastic_QBBC</td>
<td>G4HadronInelasticQBBC</td>
</tr>
<tr>
<td>g4h-phy_FTFP_BERT</td>
<td>HadronPhysicsFTFP_BERT</td>
</tr>
<tr>
<td>g4h-phy_FTFP_BERT_HP</td>
<td>HadronPhysicsFTFP_BERT_HP</td>
</tr>
<tr>
<td>g4h-phy_FTFP_BERT_TRV</td>
<td>HadronPhysicsFTFP_BERT_TRV</td>
</tr>
<tr>
<td>g4h-phy_FTF_BIC</td>
<td>HadronPhysicsFTF_BIC</td>
</tr>
<tr>
<td>g4h-phy_QGSP_BERT</td>
<td>HadronPhysicsQGSP_BERT</td>
</tr>
<tr>
<td>g4h-phy_QGSP_BERT_HP</td>
<td>HadronPhysicsQGSP_BERT_HP</td>
</tr>
<tr>
<td>g4h-phy_QGSP_BIC</td>
<td>HadronPhysicsQGSP_BIC</td>
</tr>
<tr>
<td>g4h-phy_QGSP_BIC_HP</td>
<td>HadronPhysicsQGSP_BIC_HP</td>
</tr>
<tr>
<td>g4h-phy_QGSP_FTFP_BERT</td>
<td>HadronPhysicsQGSP_FTFP_BERT</td>
</tr>
<tr>
<td>g4h-phy_QGS_BIC</td>
<td>HadronPhysicsQGS_BIC</td>
</tr>
<tr>
<td>g4h-phy_Shielding</td>
<td>HadronPhysicsShielding</td>
</tr>
<tr>
<td>g4ion-binarycascade</td>
<td>G4IonBinaryCascadePhysics</td>
</tr>
<tr>
<td>g4ion-inclxx</td>
<td>G4IonINCLXXPhysics</td>
</tr>
<tr>
<td>g4ion</td>
<td>G4IonPhysics</td>
</tr>
<tr>
<td>g4ion-QMD</td>
<td>G4IonQMDPhysics</td>
</tr>
<tr>
<td>g4n-trackingcut</td>
<td>G4NeutronTrackingCut</td>
</tr>
<tr>
<td>g4optical</td>
<td>G4OpticalPhysics</td>
</tr>
<tr>
<td>g4radioactivedecay</td>
<td>G4RadioactiveDecayPhysics</td>
</tr>
<tr>
<td>g4stopping</td>
<td>G4StoppingPhysics</td>
</tr>
</tbody>
</table>
Reference Physics Lists

Reference physics lists are pre-made, complete lists provided by Geant4.

One complication with reference lists is that they do not support use of Parallel Worlds. This means that you cannot place components into a parallel world, and, for the Dividable Components (TsBox, TsCylinder and TsSphere), you cannot score with a different set of divisions than you have set for the component itself (we handle such complex scoring by creating parallel worlds). TOPAS will give an error if you attempt to use a reference list in a situation where parallel worlds are needed. In such situations, use Modular Physics Lists.

The names of the reference physics lists, and their detailed descriptions, are here.

To use a reference physics list, specify the list name in the Type parameter, such as:

```
Ph/Default/Type = "QGSP_BERT_HP"
```

Reference physics lists allow only one additional option:

```
Ph/Default/CutForAllParticles = 0.05 mm # single range cut to use for all particles
```

Optical Physics

Optical Photons


TOPAS allows to include optical physics by means of the g4optical module in the physics list. The available optical processes included in the g4optical module are: scintillation, Cerenkov radiation, wavelength shifting, optical absorption, Rayleigh scattering and boundary processes. However, this module is not sufficient to set up the generation and tracking of optical photons. The optical properties of the material of the volumes must to be defined too (at the least the refractive index must to be defined). There exist two types of variables to define the optical properties: a vector based and constant based. The vector-based parameter allows to define a property (refractive index for example) as a function of the photon’s energy. While the constant-based parameters allows to define an scalar (scintillation yield for example).

To activate the optical properties in a material one must to set:

```
Ma/MyMaterial/EnableOpticalProperties = "True"
```

To set a property based on a vector, one must to define the energy of reference. For example to include the refractive index one must to define two parameters:

```
Ma/MyMaterial/RefractiveIndex/Energies = 3 2.0 2.5 3.0 eV
Ma/MyMaterial/RefractiveIndex/Values = 3 1.58 1.58 1.58
```

To set a property based on a scalar only one parameter is needed, for example:

```
Ma/MyMaterial/ScintillationYield = 1120 # in ph/MeV
Ma/MyMaterial/FastTimeConstant = 2.1 ns
```

The full list of parameters available is listed in the next table.
Optical Surfaces

If a perfect smooth interface is between two dielectric materials, the user only needs to provide the refractive index. In all other cases, a surface or optical boundary needs to be defined. There exist two kinds of surfaces: the border surface that delimits the boundary between two components; and the skin surface which surrounds one single component. Border surface is ordered in the sense that the order of the components matters, two border surfaces can exists between a pair of components. Thus, the follow parameters define two surfaces for a pair of components:

\[
\begin{align*}
  & s: Ge/MyComponent1/OpticalBehaviorTo/MyComponent2 = "MySurface1" \\
  & s: Ge/MyComponent2/OpticalBehaviorTo/MyComponent1 = "MySurface2"
\end{align*}
\]

For skin surface only one surface can be defined per component:

\[
  & s: Ge/MyComponent1/OpticalBehavior = "MySurface1"
\]

Surfaces can be defined as follows (see next table for description):

\[
  & s: Su/MySurfaceName/Type = "dielectric_dielectric" \# or dielectric_metal
\]

Next, choose the model for optical surfaces:

\[
  & s: Su/MySurfaceName/Model = "Glisur " \# Or Unified
\]

Finally the finish:

\[
  & s: Su/MySurfaceName/Finish = "Polished"
\]
In addition, more detailed properties can be added by parameters described in the table below. In such a case, the way to define would be for example (with prefix Su instead of Ma):

```
dv:Su/MySurfaceName/Energies = 2 1.0 4.0 eV
uv:Su/MySurfaceName/Reflectivity = 2 0.8 0.8
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Parameter name</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>Type</td>
<td>dielectric_dielectric</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dielectric_metal</td>
</tr>
<tr>
<td>string</td>
<td>Finish</td>
<td>polished: smooth perfectly polished surface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>polishedfrontpainted: smooth top-layer (front) paint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>polishedbackpainted: same as polished but with a back-paint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ground: rough surface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>groundfrontpainted: rough top-layer (front) paint</td>
</tr>
<tr>
<td></td>
<td></td>
<td>groundbackpainted: same as ground but with a back-paint</td>
</tr>
<tr>
<td>string</td>
<td>Model</td>
<td>Unified: reference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Glisur: original GEANT3.21 model</td>
</tr>
<tr>
<td>unitless</td>
<td>SigmaAlpha</td>
<td>Between 0 and 1. By default 0</td>
</tr>
</tbody>
</table>

### Miscellaneous

#### User-Supplied Physics Lists

See *Custom Physics Lists* for details on how to provide your own physics list. This option is not recommended unless you have significant Geant4 expertise.

#### Multiple Physics Lists

You can have more than one list defined at the same time, but only the one specified in `Ph/ListName` will actually be in effect:

```
s:Ph/ListName = "MyList1"
s:Ph/MyList1/Type= "QGSP_BERT_HP" # This list is in effect now
d:Ph/MyList1/CutForAllParticles = 0.05 mm
...
s:Ph/MyList2/Type= "Geant4_Modular" # This list goes into effect if Ph/ListName set into MyList2
sv:Ph/MyList2/Modules = 1 "g4em-standard_opt3"
d:Ph/MyList2/CutForGamma = 0.04 mm
```
Production Thresholds

Production Thresholds and range cuts are discussed in detail in the Geant4 Application Developers Guide. By default, appropriate limits are set by the physics list. You can override these defaults with:

```
d:Ph/MyPhysics/SetProductionCutLowerEdge = 200 eV
d:Ph/MyPhysics/SetProductionCutHighEdge = 30 MeV
```

Step Size

The selection of step size is a complex issue in Monte Carlo tracking. Geant4 has its own complex logic for automatically selecting what it thinks will be an appropriate step size, based on local geometry and physics, and the user will not generally need to override this automatic behavior. However, your applications may be sensitive to this behavior, and you may therefore want to set a maximum step size in certain components. In general, larger step sizes give faster performance, but smaller step sizes may give better accuracy.

To limit Geant4’s maximum step size in a given component:

```
d:Ge/MyComponent/MaxStepSize = 1. mm # sets maximum step size used in this component
```

Step size settings do not affect other Components placed within this Component. You must explicitly set the step size for any subcomponents that you want to affect.

The choice of maximum step size is highly dependent on your exact simulation problem. If you think you need to set a maximum step size, try running with several values, and pick one for which a small variation up or down does not cause a significant change in results.
Introduction

There are two basic classes of scorers:

- **Volume Scorers** (e.g. Energy or Dose)
- **Surface Scorers** (e.g. Track Count or Phase Space)

Most scorers output overall quantities that are accumulated over many particles (counts and averages), but other scorers can output specific information per particle (in an n-tuple format).

You can have any number of scorers. A scorer is defined when you have a line that ends with `Quantity`, such as:

```
s:Sc/MyScorer/Quantity = "DoseToMedium"
```

You may write your own additional scorers (see *Custom Scorers*).

### Volume Scorers

Here are the available volume scorers:

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoseToMedium</td>
<td>sum of energy deposits divided by mass</td>
</tr>
<tr>
<td>DoseToWater</td>
<td>from energy-dependent stopping power conversion (see below)</td>
</tr>
<tr>
<td>DoseToMaterial</td>
<td>from energy-dependent stopping power conversion (see below)</td>
</tr>
<tr>
<td>EnergyDeposit</td>
<td>sum of step lengths divided by volume</td>
</tr>
<tr>
<td>Fluence</td>
<td>sum of step lengths times energy divided by volume</td>
</tr>
<tr>
<td>EnergyFluence</td>
<td>counting method described below</td>
</tr>
<tr>
<td>Charge</td>
<td>counting method described below</td>
</tr>
<tr>
<td>EffectiveCharge</td>
<td></td>
</tr>
<tr>
<td>StepCount</td>
<td></td>
</tr>
<tr>
<td>ProtonLET</td>
<td>various methods described below</td>
</tr>
</tbody>
</table>
Volume Scorers must indicate the relevant Component:

```plaintext
a:Sc/MyScorer/Component = "Phantom"
```

For DoseToMaterial, you must also specify the Material:

```plaintext
a:Sc/MyScorer/Material = "SomeMaterial"
```

Note that in this case, the material name must exactly match the case defined in Geant4. To check what materials have been defined, add the parameter:

```plaintext
i:Ma/Verbosity = 1
```

For DoseToWater and DoseToMaterial, we use energy-dependent stopping power conversion as in:

```plaintext
dose_to_new_material = dose_to_medium * (density_of_medium / density_of_new_material) * (dEdX_in_new_material / dEdX_in_medium)
```

The \(dEdX\) comes from the Geant4 EmCalculator utility.

The DoseToWater and DoseToMaterial scorers are somewhat slow since, for every hit, they need to compute stopping power ratios based on the current energy of the particle. You can obtain better speed by adding the option:

```plaintext
b:Sc/MyScorer/PreCalculateStoppingPowerRatios = "True" # defaults to "False"
```

- False gives the best accuracy, calculating stopping power on-the-fly for the exact energy.
- True gives the best speed, looking up stopping power from a pre-calculated table binned by energy. It is about 50% faster than the default option for typical patient simulations. The difference in accuracy is not significant for most studies.

For PreCalculateStoppingPowerRatios, the table of stopping power ratios can be tuned by:

```plaintext
Sc/MyScorer/ProtonEnergyBinSize # default is 1 MeV
Sc/MyScorer/MinProtonEnergyForStoppingPowerRatio # default is 1 MeV
Sc/MyScorer/MaxProtonEnergyForStoppingPowerRatio # default is 500 MeV
Sc/MyScorer/ElectronEnergyBinSize # default is 1 keV
Sc/MyScorer/MinElectronEnergyForStoppingPowerRatio # default is 1 keV
Sc/MyScorer/MaxElectronEnergyForStoppingPowerRatio # default is 1 MeV
```

For Charge and EffectiveCharge:

- If a particle reaches zero kinetic energy in the scoring volume, its charge is accumulated
- If a particle is generated in the scoring volume, its charge is subtracted

**ProtonLET Scorer**

The ProtonLET scorer gives the LET of primary and secondary protons, including the energy deposited by associated secondary electrons. It uses techniques discussed in two recent articles on best practices to score LET in Geant4:


In particular, we adopt the methods developed by Granville and Sawakuchi. We compute dose-averaged LET, but you may instead request track-averaged:
s:Sc/MyScorer/WeightBy = "Track" # defaults to "Dose"

By default, the LET is computed by dividing the energy deposited by the step length. Such distributions can feature spurious spikes, caused by events where the step length is severely constrained by a voxel boundary crossing. Three solutions to this issue are provided:

- By default, a step-by-step upper cut-off is set, such that steps contributing greater than this value are not be scored:
  \[ d:Sc/MyScorer/MaxScoredLET = 100 \text{ MeV/mm}/(\text{g/cm}^3) \] # default 100 MeV/mm/(g/cm³)

- Alternately, you can set the LET computation to look up the electronic stopping power for the pre-step proton energy:
  \[ b:Sc/MyScorer/UsePreStepLookup = "True" \] # defaults to "False"

- Or you can increase the electron production threshold:
  \[ d:Ph/Default/CutForElectron = 1 \text{ mm} \] # defaults to 0.05 mm

The ProtonLET Scorer can give values that are too high in air, where the mean path length between discrete processes can be larger than the voxel size. This can be avoided by neglecting secondary electrons, with:

\[ d:Sc/MyScorer/NeglectSecondariesBelowDensity = 0.1 \text{ g/cm}^3 \]

Even when you do this, rare events that produce very low energy protons (e.g. a recoiling hydrogen nucleus) will produce spikes in LET. This is also seen in the PreStepLookup version of the scorer. They are not seen in the fluence-averaged version of the scorer, since they are rare events. For this reason we introduce the parameter:

\[ d:Sc/MyScorer/UseFluenceWeightedBelowDensity = 0. \text{ g/cm}^3 \]

We set this to zero by default because it is strange to mix both types of LET in a single distribution, and could be significantly wrong at the end of range. We expect users to want to enable this when making a pretty plot of LET to overlay on a CT scan, without spikes in cavities and outside the patient.

### Surface Scorers

Surface Scorer Quantities are:

- SurfaceCurrent
- SurfaceTrackCount
- PhaseSpace

Surface Scorers must indicated the relevant Component and Surface name:

\[ s:Sc/MyScorer/Surface = "Phantom/ZMinusSurface" \]

where the surface name refers to the coordinate system of the Component.

The syntax to specify surface depends on which shape component is involved.

- For TsBox:
  - XMinusSurface
  - XPlusSurface
- YMinusSurface
- YPlusSurface
- ZMinusSurface
- ZPlusSurface

• For TsCylinder:
  - ZMinusSurface
  - ZPlusSurface
  - InnerCurvedSurface
  - OuterCurvedSurface
  - PhiMinusSurface (if cut or divided along Phi)
  - PhiPlusSurface (if cut or divided along Phi)

• For TsSphere:
  - InnerCurvedSurface
  - OuterCurvedSurface
  - PhiMinusSurface (if cut or divided along Phi)
  - PhiPlusSurface (if cut or divided along Phi)
  - ThetaMinusSurface (if cut or divided along Theta)
  - ThetaMinusSurface (if cut or divided along Theta)

If you are scoring on a divided component (TsBox, TsCylinder or TsSphere), all surfaces of the divided component then become sensitive for scoring. So, for example, ZMinusSurface will mean to accumulate hits on every ZMinusSurface of every voxel in the divided TsBox.

Creators of parameter files can pre-define more user-friendly synonyms through relative parameters, such as:

```plaintext
d:Ge/WaterTank/Water/UpstreamSurface = Ge/WaterTank/Water/ZMinusSurface
```

so that users can then score using the named `Surface`, as in:

```plaintext
d:Sc/MyScorer/Surface = Ge/WaterTank/Water/UpstreamSurface
```

**Phase Space Scorer**

Phase Space refers to the technique of saving or replaying a set of particles crossing a given surface. It is the only one of our built-in scorers that saves data to n-tuple format, rather than storing accumulated overall data (counts or averages). However you can write extension scorers to use this generalized n-tuple capability to store other information on a per-particle basis (see *Custom Scorers*).

- When one saves a phase space, one defines a surface and then saves the position, particle type, energy and momentum of some or all particles crossing that surface.
- When one replays a phase space, one starts a set of particles from the saved positions, with the saved particle types, energy and momentum.
Phase Space enables separating two parts of a simulation or analysis job, and can be used to transfer sets of particles among different codes.

If your **Surface Scorer** has **Quantity** = "PhaseSpace", the output will be a pair of Phase Space files:

- A .header file tells the number of histories, the number of saved particles and the order of information in the .phsp file
- A .phsp file contains all the details of all the saved particles

We support three formats for Phase Space:

- **ASCII** provides particle information in an easy to read simple text file, which data encoded as a series of columns of text. The header file tells the contents and column order per particle.
- **Binary** provides the same information as ASCII, but in a much more compact format, with data encoded in a stream of bytes. The header file tells the contents and byte order per particle. Use Binary in cases where the ASCII format produces excessively large files.
- **Limited** is an alternate binary format compatible with some legacy codes. It has fewer options for what data can be expressed, but is compatible with codes such as that used by Varian for their TrueBeam phase space files. Use Limited format only when you need to exchange phase space with legacy codes.

You can additionally write phase space to **ROOT files**, however there is no corresponding ability to read phase space back in from these files.

You tell TOPAS what format to write out by setting:

```plaintext
s:Sc/MyScorer/OutputType = "ASCII" # "Binary", "ASCII," "Limited" or "ROOT"
```

All formats provide at least ten quantities for each scored particle:

- X position
- Y position
- Z position
- U (direction cosine of momentum with respect to X)
- V (direction cosine of momentum with respect to Y)
- Energy in MeV
- Weight
- Particle ID
- Flag to tell if Third Direction Cosine is Negative (1 means true)
- Flag to tell if this is the First Scored Particle from this History (1 means true) (Note that this may or may not be the primary, as the primary may or may not have made it all the way to the scoring plane).

The positions are relative to the center of the **World**.

For the ASCII and Binary formats, you can turn on additional columns of phase space output:

```plaintext
b:Sc/MyScorer/IncludeTOPASTime = "True" # Time used by TimeFeatures for this history
b:Sc/MyScorer/IncludeTimeOfFlight = "True" # Time of Flight of this particle from start of history to scoring plane
b:Sc/MyScorer/IncludeRunID = "True"
```
```plaintext
b:Sc/MyScorer/IncludeEventID = "True"
```
```plaintext
b:Sc/MyScorer/IncludeTrackID = "True"
```
```plaintext
b:Sc/MyScorer/IncludeParentID = "True" # Track ID of parent particle
```
```plaintext
b:Sc/MyScorer/IncludeCharge = "True"
```
The last of these gives the four variable parts of a random seed. Replaying this random seed will get you the same event back later. The full random seed should be a file of the form:

```
Uvec
1878463799
3
1425618182
1466214412
```

To reuse a saved seed, create a file with the above five lines, replacing the four numeric parts with the four integers in the phase space file. Assuming you name that file `event1.rndm`, you can then make TOPAS start from this random seed by having TOPAS wake up at the Geant4 command line, by using:

```
Ts/PauseBeforeSequence = "True"
```

And then typing:

```
/random/resetEngineFrom event1.rndm
exit
```

The phase space scorer and any custom n-tuple scorers buffer output to avoid excessive disk access. You will not generally need to adjust this buffering value, but can adjust if you wish:

```
i:Sc/MyScorer/OutputBufferSize = 1000 # Number of particles in phase space buffer
```

### Phase Space Format

Phase Space refers to the technique of saving or replaying a set of particles crossing a given surface.

- When one saves a phase space, one defines a surface and then saves the position, particle type, energy and momentum of some or all particles crossing that surface.
- When one replays a phase space, one starts a set of particles from the saved positions, with the saved particle types, energy and momentum.

Phase Space enables separating two parts of a simulation or analysis job, and can be used to transfer sets of particles among different codes.

A Phase Space is stored as a pair of related files:

- A `.header` file tells the number of histories, the number of saved particles and the order of information in the `.phsp` file
- A `.phsp` file contains all the details of all the saved particles

We support three formats for Phase Space:

- Binary is a compact format, with data encoded in a stream of bytes. The header file tells the contents and byte order per particle.
- ASCII provides the same information as Binary, but presents it as a much less compact, but easier to read simple text file, which data encoded as a series of columns of text. The header file tells the contents and column order per particle.
- Limited is an alternate binary format compatible with some legacy codes. It has fewer options for what data can be expressed, but is compatible with codes such as that used by Varian for their TrueBeam phase space files.
You can additionally write phase space to **ROOT files**, however there is no corresponding ability to read phase space back in from these files.

For the Binary and ASCII formats, Particle ID is encoded using the large set of integer codes specified by the Particle Data Group (PDG):

- 22 = photon
- 11 = electron
- -11 = positron
- 2112 = neutron
- 2212 = proton
- Additional codes go all the way up to ten digit ion codes of the form ±10LZZZAAA.
- See the PDG web site for a full explanation

For the Limited format, only a few particle codes are supported, while other particle types are not scored at all (and so this format is only recommended if you need to interface with legacy codes):

- 1 = photon
- 2 = electron
- 3 = positron
- 4 = neutron
- 5 = proton

The Binary and ASCII formats are self-describing, with the complete column or byte order documented in the associated header file. The exact set of columns will depend on which options are used to create the phase space file. Run the `WriteASCII.txt` and `WriteBinary.txt` examples to see these headers.

If you are attempting to create TOPAS Binary or ASCII phase space from some application other than TOPAS, be advised that the formatting requirements are very specific. It is best to compare your phase space header and phsp files to those produced by the TOPAS examples listed above. Some things to watch out for:

- First line of header has to be exactly as produced by TOPAS, with no extra spaces, tabs, etc.
- Integer values in the ASCII phase space must not contain decimal points

The Limited format uses the following byte order (the format is not self-describing):
Particle ID
Absolute value gives the particle code
Sign of this value encodes the direction of the 3rd direction cosine

Energy
Absolute value gives the energy in MeV
Sign of this value is set to negative if this is the first scored particle from this history

X position
Y position
Z position
U (direction cosine of momentum with respect to X)
V (direction cosine of momentum with respect to Y)
Weight

Direction cosines are consistent between Binary, ASCII and Limited formats. Descriptions can be found on Wikipedia and on MathWorld. Direction cosines U, V and W correspond to direction cosines alpha, beta and gamma on those sites.

Filtering Scorers

You may add filters to limit what is scored.
You may write your own additional filters (see Custom Filters).
Filter by Generation. Accepts either "Primary" or "Secondary":

```plaintext
@Sc/MyScorer/OnlyIncludeParticlesOfGeneration = "Primary"
```

Filter by Charge. Accepts one or more of "Positive", "Negative" or "Neutral":

```plaintext
@v:Sc/MyScorer/OnlyIncludeParticlesCharged = 1 "Negative"
@v:Sc/MyScorer/OnlyIncludeParticlesNotCharged = 1 "Negative"
```

Filter by Atomic Mass or Number:

```plaintext
i:Sc/MyScorer/OnlyIncludeParticlesOfAtomicMass = 10  # allow all ions of atomic mass 10
i:Sc/MyScorer/OnlyIncludeParticlesNotOfAtomicMass = 10
i:Sc/MyScorer/OnlyIncludeParticlesOfAtomicNumber = 6   # allow all ions of Carbon
i:Sc/MyScorer/OnlyIncludeParticlesNotOfAtomicNumber = 6
```

Filter by Particle’s Initial Kinetic Energy:

```plaintext
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKEBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKENotBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKE = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKENot = 1. MeV
```
When designing energy or momentum filters, keep in mind that since no vacuum is perfect in Geant4 (density can be low but cannot be exactly zero), even particles traveling through "Vacuum" will experience some energy loss.

Filter by Particle’s Initial Momentum:

```
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentumBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentumNotBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentum = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentumNot = 1. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentumAbove = 10. MeV
d:Sc/MyScorer/OnlyIncludeParticlesWithInitialMomentumNotAbove = 10. MeV
```

Filter by Kinetic Energy of Particle or its Ancestor when it hit the Scoring Component (excludes any particles descended from primaries that originated in the component):

```
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKEBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKENotBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKE = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKENot = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKEAbove = 10. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKENotAbove = 10. MeV
```

Filter by Initial Momentum of Particle or its Ancestor when it hit the Scoring Component (excludes any particles descended from primaries that originated in the component):

```
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentumBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentumNotBelow = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentum = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentumNot = 1. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentumAbove = 10. MeV
d:Sc/MyScorer/OnlyIncludeIfIncidentParticleMomentumNotAbove = 10. MeV
```

Filter by Process that created the particle. Allows one or more process name:

```
sv:Sc/MyScorer/OnlyIncludeParticlesFromProcess = 2 "hIoni" "eBrem"
sv:Sc/MyScorer/OnlyIncludeParticlesNotFromProcess = 2 "hIoni" "eBrem"
```

Filter by Process that created the particle or any of its ancestors:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorFromProcess = 2 "hIoni" "eBrem"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotFromProcess = 2 "hIoni" "eBrem"
```

Filter by Particle Name:

```
sv:Sc/MyScorer/OnlyIncludeParticlesNamed = 2 "proton" "neutron"
sv:Sc/MyScorer/OnlyIncludeParticlesNotNamed = 2 "proton" "neutron"
```

Filter by Particle Name or the name of any of the particle’s ancestors. Use this to, for example, score all charge that results from neutrons, even if the final particle is not a neutron:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNamed = 1 "neutron"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotNamed = 1 "neutron"
```

Particle names are as described [here](#).
Filter by Particle’s Origin Volume, Component, or Component and Subcomponents:

```
sv:Sc/MyScorer/OnlyIncludeParticlesFromVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeParticlesNotFromVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeParticlesFromComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeParticlesNotFromComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeParticlesFromComponentOrSubComponentsOf = 1 "Nozzle"
sv:Sc/MyScorer/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf = 1 "Nozzle"
```

Filter by Particle or its Ancestor’s Origin Volume, Component, or Component and Subcomponents:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorFromVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotFromVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorFromComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotFromComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf = 1 "Nozzle"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf = 1 "Nozzle"
```

Filter by whether Particle Interacted in Volume, Component, or Component and Subcomponents:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleInteractedInVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotInteractedInVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleInteractedInComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotInteractedInComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf = 1 "Nozzle"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf = 1 "Nozzle"
```

Filter by Particle or its Ancestor Interacted in Volume, Component, or Component and Subcomponents:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorInteractedInVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorInteractedInComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf = 1 "Nozzle"
sv:Sc/MyScorer/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf = 1 "Nozzle"
```

Filter by whether Particle Traversed Volume, Component, or Component and Subcomponents:

```
sv:Sc/MyScorer/OnlyIncludeIfParticleTraversedVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotTraversedVolume = 1 "Propeller20/Leaf"
sv:Sc/MyScorer/OnlyIncludeIfParticleTraversedComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotTraversedComponent = 1 "Jaws"
sv:Sc/MyScorer/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf = 1 "Nozzle"
sv:Sc/MyScorer/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf = 1 "Nozzle"
```

Filter by Particle or its Ancestor Traversed Volume, Component, or Component and Subcomponents:
Filter by material:

```latex
\texttt{sv:Sc/MyScorer/OnlyIncludeIfInMaterial = 2 "G4\_WATER" "Air"}
\texttt{sv:Sc/MyScorer/OnlyIncludeIfNotInMaterial = 2 "G4\_WATER" "Air"}
```

Note that in this case, the material name must exactly match the case defined in Geant4. To check what materials have been defined, add the parameter:

```latex
\texttt{i:Ma/Verbosity = 1}
```

Filter on DICOM RT Structure Sets: A structure set is an extra file in the DICOM directory that provides information on structures such as organs, tumors, PTVs, etc. that have been outlined (contoured) in the planing process. The data is stored as a set of polygons, up to one per slice for each contoured structure. TOPAS can color code DICOM components according to this structure information (see \textit{Patient in DICOM Format}) and can filter scoring based on these structures:

```latex
\texttt{sv:Sc/MyScorer/OnlyIncludeIfInRTStructure = 2 "R\_LUNG" "L\_LUNG"}
```

If the structure name includes a space, substitute an underscore in the parameter. So, for example, if the structure name is “R LUNG”, you should supply the parameter as “R\_LUNG”.

For Surface Scorers, you can also filter by whether particle is going "In" or "Out" of scoring surface. Omit this filter to allow either option:

```latex
\texttt{s:Sc/MyScorer/OnlyIncludeParticlesGoing = "In"}
```

You may specify more than one filter. For example, to score protons with initial KE over 100 MeV:

```latex
\texttt{sv:Sc/MyScorer/OnlyIncludeParticlesNamed = 1 "proton"}
\texttt{d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKEAbove = 100. MeV # minimum energy}
```

You can invert the results of all previous filters. The following would score only particles that are Not protons with initial KE over 100 MeV:

```latex
\texttt{sv:Sc/MyScorer/OnlyIncludeParticlesNamed = 2 "proton" "neutron"}
\texttt{d:Sc/MyScorer/OnlyIncludeParticlesWithInitialKEAbove = 100. MeV # minimum energy}
\texttt{b:Sc/MyScorer/InvertFilter = "True"}
```

Any filter property can be set by \textit{Time Features} if you wish, to produce time-dependent filtering.

### Output Specification

Scored quantities can be output to simple files (csv or binary formats), data files for use in analysis systems (ROOT or XML format) or to a DICOM file. There are also options to directly produce Volume Histograms (such as DVH).

To specify output file name:
Note that this can be more than just a file name - it can include a relative or absolute file path, as in:

```plaintext
s:Sc/MyScorer/OutputFile = "/myOutputFileName" # one directory above current directory
s:Sc/MyScorer/OutputFile = "/SomeSubdirectory/myOutputFileName"
```

To specify output file type for all except the *Phase Space Scorer*:

```plaintext
s:Sc/MyScorer/OutputType = "csv" # "csv", "binary", "Root", "Xml" or "DICOM"
```

For binary output of 3D data, such as from scoring in a water phantom or a patient, the following table shows the correspondence between TOPAS divisions and common 3D data viewing applications:

<table>
<thead>
<tr>
<th>TOPAS</th>
<th>fNi (X/R/R)</th>
<th>fNj (Y/Phi/Phi)</th>
<th>fNk (Z/Theta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageJ</td>
<td>Width</td>
<td>Height</td>
<td>Images</td>
</tr>
<tr>
<td>ParaView</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Amide</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td>Numpy</td>
<td>Use the python module <code>topas2numpy</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DICOM export is handled through the package *GDCM*, which is pre-built into TOPAS.

DICOM output is in the form of a single `.dcm` file that contains DICOM header information (voxel size, spacing, etc.) and then a block of binary image data representing the 3D output. We use the DICOM output format called `MONOCHROME2`. Each pixel is represented by a 16 or 32 bit integer.

- 16 bit is the default.
- For 32 bit, specify:

```plaintext
b:Sc/MyScorer/DICOMOutput32BitsPerPixel = "True"
```

Because TOPAS accumulates all scores as doubles, you must tell TOPAS how to convert these double values to the DICOM 16bit or 32bit integers. You do so by providing a scaling factor:

```plaintext
u:Sc/MyScorer/DICOMOutputScaleFactor = 1.0E12
```

The appropriate value depends on your specific setup. You need a factor that will make your expected double values fit into an appropriate range of 16bit or 32bit integers:

- 16bit integers are 0 through 65,536
- 32bit integers are 0 through 4,294,967,296

- For example, if you expect to score values that are only on the order of 1.0E-9, and you are using 16bit, an output scale factor of 1.0E12 will give you reasonable integers on order of 1000

“Root” and “Xml” will generate histogram files. Specify the binning of the scored quantity as follows:

```plaintext
i:Sc/MyScorer/HistogramBins = 100 # number of bins
d:Sc/MyScorer/HistogramMin = 0, MeV # with unit appropriate to scored quantity
d:Sc/MyScorer/HistogramMax = 100, MeV # with unit appropriate to scored quantity
```

Histograms will be either 1D or 2D depending on how the scoring geometry is divided or the energy is binned (see *Binning by Energy*).

- If the geometry is undivided and there is no energy binning, a 1D histogram is produced.
- If the geometry is undivided and there is energy binning, the second histogram axes will be energy.
• If the geometry is divided, it can only be divided in one dimension (such as either X, Y or Z for TsBox geometries) and there can be no energy binning. The second histogram axes will be the axes of the geometry division.

All histogram output is combined into a single file, such as `topas.root` or `topas.xml`. The histogram file name can be adjusted by:

```
s:Sc/RootFileName = "topas" # name for ROOT output file
s:Sc/XmlFileName = "topas" # name for XML output file
```

Physicists often report the quality of a treatment plan by showing Dose Volume Histograms (DVHs). Such histograms represent what fractional volume of a given structure has received a given Dose.

• In a differential DVH, the bin value indicates what percentage of the structure volume received the given dose.
• In a cumulative DVH, the bin value indicates what percentage of the structure volume received at least the given dose (the zeroth bin will always have a value of 1, since all bins receive at least zero dose).
• If you combine TOPAS DVH options with the filtering option `OnlyIncludeIfInRTStructure` (see Filtering Scorers), you can generate a DVH for a specific contoured structure (such as DVH to R_LUNG).

TOPAS can generate a Volume Histogram for any scored quantity, not just Dose. Just set the scorer’s report parameter to include either "DifferentialVolumeHistogram" or "CumulativeVolumeHistogram", as in:

```
sv:Sc/DoseAtPhantom/Report = 1 "CumulativeVolumeHistogram"
```

You cannot specify both types of volume histograms in a single scorer, but you can specify other reporting options, such as the following, which will give a basic histogram of "Sum" and "Mean", plus a "CumulativeVolumeHistogram":

```
sv:Sc/DoseAtPhantom/Report = 3 "Sum" "Mean" "CumulativeVolumeHistogram"
```

As with any histogram, you also need to specify `HistogramBins`, `HistogramMin` and `HistogramMax`. For an example, see `DoseVolumeHistogram.txt`.

If your results has a 1 in the first bin and zero in the other bins, it probably means your `HistogramMax` was set too high, and thus none of the voxels had enough dose to get beyond the zeroth bin.

To specify output file type for the phase space scorer:

```
x:Sc/MyScorer/OutputType = "ASCII" # "ASCII" or "Binary"
```

ASCII format has the advantage that it is human-readable text. Binary format has the advantage that it is much more compact, hence suitable for large files.

By default, output will occur just once, after the entire session. But if you are using Time Features and wish to have separate output from specific runs:

```
b:Sc/MyScorer/OutputAfterRun = "True" # set True to trigger output of scorer after this run
# If this is always set False, or not defined, we just output at the end of the session.
# If this is always set True, we output after every run.
```

Additional output control options:

```
b:Sc/MyScorer/OutputToConsole = "True" # control whether output is also dumped to console
s:Sc/MyScorer/IfOutputFileAlreadyExists = "Increment" # "Exit", "Overwrite" or "Increment"
```
We keep name and type separate in the above so that one can do things like change all output from csv to AIDA by just changing a single parameter (by setting many `OutputType` parameters equal to a common relative string parameter).

**Miscellaneous**

### Binning in Dividable Components

When scoring in *Dividable Components* (TsBox, TsCylinder or TsSphere), you have many binning options. By default, binning will match the divisions of the volume. So if you have divided the component, the score will be divided in the same manner.

You are also free to specify some other binning.

- In a TsBox, you can specify binning in X, Y and Z:

```plaintext
i:Sc/MyScorer/XBins = 512
i:Sc/MyScorer/YBins = 512
i:Sc/MyScorer/ZBins = 256
```

- In a TsCylinder, you can specify binning in R, Phi and Z:

```plaintext
i:Sc/MyScorer/RBins = 100
i:Sc/MyScorer/PhiBins = 20
i:Sc/MyScorer/ZBins = 1
```

- In a TsSphere, you can specify binning in R, Phi and Theta:

```plaintext
i:Sc/MyScorer/RBins = 20
i:Sc/MyScorer/PhiBins = 20
i:Sc/MyScorer/ThetaBins = 1
```

Behind the scenes, TOPAS uses Geant4’s parallel worlds system to support this binning flexibility. When scoring binning is different from the component’s natural binning, TOPAS actually scores in a parallel world copy of the component. This is all done automatically.

### Binning by Energy

Any scorer can be binned by incident particle kinetic energy, the energy of the particle or its ancestor when it first hit the scoring component, by including the following:

```plaintext
i:Sc/MyScorer/EBins = 10 # defaults to 1, that is, un-binned
d:Sc/MyScorer/EBinMin = 0. MeV # defaults to zero
d:Sc/MyScorer/EBinMax = 100. MeV # must be specified if EBins is greater than 1
```

The output will include three extra bins, one for underflow (energy < `EBinMin`), one for overflow (energy > `EBinMax`) and one for the case where there is no incident track (the primary particle was created already inside the scoring component, so it was never incident upon the scoring component).

### Binning by Time

Any scorer can be binned by time-of-flight, the elapsed time since the history was generated (in Geant4 this is called “global time”):
The output will include two extra bins, one for underflow (time < TimeBinMin) and one for overflow (time > TimeBinMax). Note that this time-of-flight is not the same as the TOPAS time feature time. To split results based on that TOPAS time, see Splitting by Time Feature.

When radioactive decay is present, some very large times can occur, as decay may be delayed for hours or days. Thus it is not unusual to have some times exceed the TimeBinMax. To get an interesting report on what particles and processes exceed TimeBinMax, set Ts/TrackingVerbosity > 0.

**Splitting by Time Feature**

To split a scorer into separate scorers depending on the current value of any selected Time Feature:

```plaintext
s:Sc/MyScorer/SplitByTimeFeature = some_time_feature_name
```

If the time feature is a Step function, one split scorer is made for each of the time feature’s values. If the time feature is a Continuous function, another parameter is expected to specify split values. This will be either a dimensioned double vector, unitless vector or integer vector, depending on the type of controlling time feature, such as:

```plaintext
dv:Sc/DoseAtPhantom/SplitByTimeFeatureValues = 5 0. 90. 180. 270. 360. deg
```

**Example 1 - Splitting under control of a Step Time Feature**

To split up a 4D CT simulation’s dose output depending on the CT time slice, where the CT time slice is controlled by:

```plaintext
s:Tf/ImageName/Function = "Step"
dv:Tf/ImageName/Values = 3 "image1" "image2" "image3"
```

The following will make the scorer DoseAtPhantom split by current value of Tf/ImageName/Value:

```plaintext
s:Sc/DoseAtPhantom/SplitByTimeFeature = "ImageName"
```

creating one scorer for each value of ImageName:

```
Sc/DoseAtPhantom-image1
Sc/DoseAtPhantom-image2
Sc/DoseAtPhantom-image3
```

**Example 2 - Splitting under control of a Continuous Time Feature**

To split up a simulation’s dose output depending on the position of a propeller, where the propeller position is controlled by:

```plaintext
s:Tf/PropellerRotation/Function = "Linear deg"
```

The following will make DoseAtPhantom split by current value of Tf/PropellerRotation/Value:

```plaintext
s:Sc/DoseAtPhantom/SplitByTimeFeature = "PropellerRotation"
dv:Sc/DoseAtPhantom/SplitByTimeFeatureValues = 5 0. 90. 180. 270. 360. deg
```

creating one scorer for each defined range of PropellerRotation:
See the `SplitByTimeFeature.txt` and `Dose4DCT.txt` examples.

**Statistical Information**

By default, scorers will report the sum of the scored quantity over all histories, but many additional reporting options are available:

```plaintext
sv:Sc/MyScorer/Report = 1 "Sum" # One or more of Sum, Mean, Histories, Count_In_Bin, Second_Moment, Variance, Standard_Deviation, Min, Max
```

Output columns will be in the same order as the values in the `Report` parameter.

When there is binning by energy or time, and there is more than one `Report` option (such as "Sum" and "Mean"), the output will be ordered as:

- Sum (underflow), Mean (underflow), Sum (bin 1), Mean (bin 1), Sum (bin 2), Mean (bin 2), etc.

"Histories" is the total number of histories that were simulated while this scorer was active (that is, excludes any histories that were produced when this scorer was gated to inactive).

"Count_In_Bin" is the number of histories that contributed to this bin (that is, excludes any histories for which no particles hit this bin).


```plaintext
for x in data:
    n = n+1
    delta = x - mean
    mean = mean + delta/n
    M2 = M2 + delta*(x - mean)
    sum = n * mean
    variance = M2/(n - 1)
    standard deviation = sqrt(variance)
```

Note that if your geometry has many divisions (such as the 70M voxels of a 512 x 512 x 256 CT), and you ask for "Mean", "Second_Moment", "Variance" or "Standard_Deviation", you will see a speed penalty. This occurs because any bin that has ever been hit will then have to recalculate its mean or second moment to account for the new history (even if the current history doesn’t hit this bin).

TOPAS calculates the variance (and hence the standard deviation) associated with the distribution of the quantity of interest (dose, fluence, etc).

- For the standard deviation of the mean value, divide the standard deviation from TOPAS by the square root of the total number of histories.
- For the standard deviation of the sum, multiply the standard deviation from TOPAS by the square root of the total number of histories.
Change Component Color Based on Scoring

You can make TOPAS recolor a component during simulation to reflect a scored value. Using this technique, you can, for example, make a box become darker as it accumulates dose. See the Darkening.txt example.

To activate this feature:

```
@:Sc/EnergyInPhantom/ColorBy = "Sum" # sum, mean, histories, standard_deviation, min, max
```

You must then provide a list of colors, and cutoff values, such as:

```
$v$:Sc/EnergyInPhantom/ColorNames = 5 "white" "grey240" "grey220" "grey200" "grey180"
$dv$:Sc/EnergyInPhantom/ColorValues = 4 1. 1000 2000 3000 MeV
```

In the above example:

- if the total energy is from 0 to 1, the phantom will be colored "White".
- if the total energy is from 1 to 1000, the phantom will be colored "grey240".
- if the total energy is from 1000 to 2000, the phantom will be colored "grey220".
- etc.

This feature must be used in conjunction with Time Features, as the color will only update after each run. And your scorer must be set to output after each run:

```
$B$:Sc/EnergyInPhantom/OutputAfterRun = "True"
```

This technique does not currently work in the Dividable Components (TsBox, TsCylinder and TsSphere). We will add this capability in a future TOPAS release. For now it only works in simple components made of single Geant4 solids.

Toggling a Scorer Off and On

To turn off a scorer:

```
$b$:Sc/MyScorer/Active = "False" # defaults to "True"
```

This feature can be combined with boolean Time Features to produce gated scoring. If the scorer skipped any values due to being set inactive at any time, the total number of skipped values is written out at in the scoring summary.

Restoring Results from Files

TOPAS provides an option to read back scored values so that you can then redo the scoring output with different options. Set the parameter:

```
Ts/RestoreResultsFromFile = "True" # defaults to "False"
```

With this set, simulation will not be run, but instead the scored values will be restored from the output of previous TOPAS simulations. For each scorer, there must be an appropriate file to read back, specified by name and type:

```
$s$:Sc/MyScorer1/InputFile = "MySavedFileName" # match exact case
$s$:Sc/MyScorer1/InputType = "csv"
```
The file to read back in must contain the appropriate scored quantity, the appropriate binning, and sufficient information to provide the new Report options. So, for example, if you previously scored "Sum" and "Histories", you could now report "Sum", "Mean", "Histories", and a DVH.

This option can also be used to read in binary output and write out csv, or vice versa.
You may have zero, one or more graphics windows active at same time:

```
Gr/MyGraphic1/Type = "OpenGL" # OpenGL, HepRep, VRML, DAWN, RayTracer, RayTracerX
```

Note that the file-based graphics systems, HepRep, VRML and DAWN may not show any image until at least one history is run. We will revisit this issue when we move to the next Geant4 version.

HepRep files are designed to be viewed in a Java application called HepRApp. Details can be found here.

Note that graphics can be one of the slowest parts of a simulation, so should be disabled if you are running a long simulation. To disable graphics, do one of the following three things:

- Comment out all of the `Gr/*/Type` parameters
- Set all `Gr/*/Active` to “False”
- Disable graphics entirely, by setting: `b:Gr/Enable = "False"

This last option is essential if you want to run on a batch system that does not contain any OpenGL graphics drivers.

File-based graphics systems will also expect a file name:

```
Gr/MyGraphic1/FileName = "MyFileName" # Defaults to name of view (which here is MyGraphic1).
```

Will use this filename plus an \_n where n increments with each refresh. Due to limitations in Geant4, FileName only affects OpenGL and HepRep. For other cases, the file name is a fixed value, g4_ followed by a file number.

This can be more than just a file name - it can include a relative or absolute file path, as in:

```
Gr/MyGraphic1/FileName = ".../MyFileName" # one directory above current directory
Gr/MyGraphic1/FileName = "/SomeSubdirectory/MyFileName"
```

Basic options:

```
b:Gr/MyGraphic1/IncludeGeometry = "True" # defaults to "True"
b:Gr/MyGraphic1/IncludeTrajectories = "True" # defaults to "True"
b:Gr/MyGraphic1/IncludeStepPoints = "True" # Show trajectory step points, defaults to "False"
```
Colors are defined by specifying their red, green, blue components, each on a scale of 0 to 255, as in:

```
iv:Gr/Color/lightblue = 3 175 255 255
```

By default, trajectories will be drawn as what Geant4 calls “Smooth Trajectories”, which means they include additional points to make them curve smoothly in a magnetic field. Geant4 does not actually use these “auxiliary points” in its simulation results, they are just present to make visualization in a field look better. In some cases, Geant4 has trouble handling these auxiliary points, and reports:

```
```
```
```
```
```

To work around this, turn off trajectory drawing or tell Geant4 not to making the trajectories smooth:

```
:b:Gr/MyGraphic1/UseSmoothTrajectories = "False" # defaults to "True"
```

You can add axes to the display. Axes lines are colored red for X, green for Y, blue for Z:

```
:b:Gr/MyGraphic1/IncludeAxes = "True" # defaults to "False"
:Gr/MyGraphic1/AxesComponent = "World" # Component in which to center the axes.
:d:Gr/MyGraphic1/AxesSize = 3. m # size of axes
```

Note that on most OpenGL graphics systems, the shadowing on the arrowheads allows you to tell whether a given axis is coming towards or away from you.

You can visualize magnetic fields, with field intensity and direction depicted through a set of arrows:

```
i:Gr/ViewA/MagneticFieldArrowDensity = 10
```

Use with caution. When combined with rotation seems to sometimes cause crashes in polycone drawing (involved in drawing the arrowheads).

By default, graphics views will refresh after every run. But you can change this to show each history individually or to accumulate all histories for the entire session (multiple runs). This applies globally to all graphics views:

```
s:Gr/RefreshEvery = "History" # "History", "Run" or "Session"
```

If parallel worlds are present, by default they will be visible. If you instead want to see only the main world, specify:

```
sv:Gr/MyGraphic1/VisibleWorlds = 1 "World" # "World", "All" or one or more specific world names
```

To turn off a graphic:

```
b:Gr/MyGraphic1/Active = "False" # defaults to "True"
```

Extra options used by OpenGL:

```
u:Gr/MyGraphic1/Zoom = 2. # increase to zoom in, decrease to zoom out
d:Gr/MyGraphic1/Theta = 45. deg # view angle as in /vis/viewer/set/viewpointThetaPhi
d:Gr/MyGraphic1/Phi = 45. deg # view angle as in /vis/viewer/set/viewpointThetaPhi
d:Gr/MyGraphic1/TransX = 0. # move left or right in the view window
d:Gr/MyGraphic1/TransY = 0. # move up or down in the view window
s:Gr/MyGraphic1/Projection = "Perspective" # Defaults to "Orthogonal"
s:Gr/MyGraphic1/PerspectiveAngle = 10. deg # Increase for stronger perspective effect
i:Gr/MyGraphic1/WindowSizeX = 600
i:Gr/MyGraphic1/WindowSizeY = 600
```
You can set Topas so that for OpenGL views, the view is copied to a file at the end of each run:

```
:b:Gr/MyGraphic1/CopyOpenGLToPDF = "True" # save to PDF
:b:Gr/MyGraphic1/CopyOpenGLToSVG = "True" # save to Scalable Vector Graphics
:b:Gr/MyGraphic1/CopyOpenGLToEPS = "True" # save to Encapsulated PostScript
:b:Gr/MyGraphic1/CopyOpenGLToPS = "True" # save to PostScript
```

Some views may result in one of the following warning messages from Geant4 Visualization. These messages are just informational and can be safely ignored. Note that even if you have set `Gr/RefreshEvery = "History"`, the view will only be copied to a file at the end of the run (not per history).

```
"WARNING: Viewpoint direction is very close to the up vector direction. Consider setting the up vector to obtain definable behavior."

"G4PhysicalVolumeSearchScene::FindVolume: Required volume "Phantom3_10x10x1", copy no. 0, found more than once. This function is not smart enough to distinguish identical physical volumes which have different parentage. It is tricky to specify in general. This function gives you access to the first occurrence only."
```

To create movies, Zoom, Theta, Phi, TransX, TransY, Projection and PerspectiveAngle can be controlled by `Time Features`.

Trajectory Coloring:

```
```

For `ColorBy = "Charge"`, trajectories default to red, greed, blue for negative, neutral and positive. You can override these defaults with:

```
sv:b:Gr/MyGraphic1/ColorByChargeColors = 3 "blue" "green" "red" # colors for neg, neutral, pos
```

For `ColorBy = "ParticleType"`, colors are Geant4 defaults:

<table>
<thead>
<tr>
<th>Particle Species</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>gamma</td>
<td>green</td>
</tr>
<tr>
<td>e-</td>
<td>red</td>
</tr>
<tr>
<td>e+</td>
<td>blue</td>
</tr>
<tr>
<td>pi+</td>
<td>magenta</td>
</tr>
<tr>
<td>proton</td>
<td>cyan</td>
</tr>
<tr>
<td>neutron</td>
<td>yellow</td>
</tr>
<tr>
<td>other</td>
<td>gray</td>
</tr>
</tbody>
</table>

You can override these settings with (particle names are described `here`):

```
sv:b:Gr/MyGraphic1/ColorByParticleTypeNames = 4 "e-" "gamma" "proton" "neutron" # any number of particle names
sv:b:Gr/MyGraphic1/ColorByParticleTypeColors = 4 "red" "green" "blue" "yellow" # for each particle type above. All other particles will be set to gray.
```
For **ColorBy = "OriginVolume"**, trajectories are grey unless they come from a named volume in:

```
sv:Gr/MyGraphic1/ColorByOriginVolumeNames = 1 "Propeller20/Leaf" # one or more volume
sv:Gr/MyGraphic1/ColorByOriginVolumeColors = 1 "red" # one color for each name above
```

For **ColorBy = "OriginComponent"**, trajectories are grey unless they come from a named component in:

```
sv:Gr/MyGraphic1/ColorByOriginComponentNames = 1 "jaws" # one or more component names
sv:Gr/MyGraphic1/ColorByOriginComponentColors = 1 "red" # one color for each name above
```

For **ColorBy = "ColorByOriginComponentOrSubComponentOf"**, trajectories are grey unless they come from a named component or any of its subcomponents in:

```
sv:Gr/MyGraphic1/ColorByOriginComponentNames = 1 "Nozzle" # one or more components
sv:Gr/MyGraphic1/ColorByOriginComponentColors = 1 "red" # one color for each name above
```

For **ColorBy = "Energy"**:

```
dv:Gr/MyGraphic1/ColorByEnergyRanges = 3 1. 4. 8. MeV # limits of energy ranges
dv:Gr/MyGraphic1/ColorByEnergyColors = 4 "red" "green" "blue" "yellow" # one for every energy interval that is defined by those ranges - one more value than number of ranges since includes less than first range value and greater than first range value
```

For **ColorBy = "Momentum"**:

```
dv:Gr/MyGraphic1/ColorByMomentumRanges = 3 1. 4. 8. MeV # limits of momentum ranges
dv:Gr/MyGraphic1/ColorByMomentumColors = 4 "red" "green" "blue" "yellow" # one for every energy interval that is defined by those ranges - one more value than number of ranges since includes less than first range value and greater than first range value
```

For **ColorBy = "Generation"**:

```
sv:Gr/MyGraphic1/ColorByGenerationColors = 2 "red" "green" # colors for primary and secondaries
```

For **ColorBy = "CreatorProcess"**:

```
sv:Gr/MyGraphic1/ColorByCreatorProcessNames = 5 "eBrem" "annihil" "Decay" "eIoni" "hIoni" # one or more process name
sv:Gr/MyGraphic1/ColorByCreatorProcessColors = 5 "red" "green" "blue" "yellow" "magenta" # one for every process name
```

To filter what trajectories will be in the graphics, use similar syntax to that used for **Filtering Scorers** and **Filtering Sources** (applies globally to all graphics views):

```
sv:Gr/OnlyIncludeParticlesNamed = 2 "proton" "neutron" # one or more particle names
sv:Gr/OnlyIncludeParticlesCharged = 1 "negative" # one or more "positive", "negative", or "neutral"
sv:Gr/OnlyIncludeParticlesFromVolume = 1 "Propeller20/Leaf" # one or more volume
sv:Gr/OnlyIncludeParticlesFromComponent = 1 "Jaws" # one or more component
sv:Gr/OnlyIncludeParticlesFromComponentOrSubComponentsOf = 1 "Nozzle" one or more
sv:Gr/OnlyIncludeParticlesWithInitialKEBelow = 1. MeV # maximum energy
sv:Gr/OnlyIncludeParticlesWithInitialKEAbove = 10. MeV # minimum energy
```
Note that the following three filters may cause a crash if the particle origin is at the world boundary:

```
Gr/OnlyIncludeParticlesFromVolume
Gr/OnlyIncludeParticlesFromComponent
Gr/OnlyIncludeParticlesFromComponentOrSubComponentsOf
```

We will study this issue again when we move to the next Geant4 version.

Visualization control for a specific component is done as part of the `Ge/` parameters for that component rather than in the `Gr/` parameters:

```
Ge/MyComponent/Color = "red"
Ge/MyComponent/DrawingStyle = "Solid" # "Solid", "Wireframe" or "FullWireFrame".
# FullWireFrame includes drawing of additional edge lines that Geant4 calls "soft edges"
# - on many graphics devices WireFrame and FullWireFrame give the same result
Ge/MyComponent/VisSegsPerCircle = 100 # Number of line segments to use to approximate a circle, defaults to 24. Set to a larger number if you want a smoother curve
Ge/MyComponent/Invisible = "True" # defaults to False meaning visible
```

We sometimes see error messages from visualization of the following form:

```
G4PhysicalVolumeSearchScene::FindVolume:
Required volume "PhantomCentralDose_1x1x40", copy no. 0, found more than once...
```

Such messages can be ignored. They do not affect the simulation results. We will revisit how to solve these error messages once we move to the next Geant4 version.
While the repeatability requirements of the TOPAS parameter system require that parameter definitions be well specified, there is still a need to define time-dependent behaviors (such as motion, beam current modulation, starting and stopping of scoring activities). The TOPAS Time Feature system allows such time-dependence to be specified in a manner that is both flexible and repeatable.

A Time Feature is a set of parameters that ultimately describes the change of a time feature $\text{Value}$. You provide parameters that define the time function, such as a linear change over time. TOPAS automatically creates a $\text{Value}$ parameter for this function (a parameter you don’t define). TOPAS continually updates this $\text{Value}$ parameter to the appropriate value for a given time.

**Note:** If you’re doing complex things with parameter file chains, you may want to know where in chain this automatically added $\text{Tf/.../Value}$ parameter goes: the answer is that it goes into the same virtual file as the $\text{Tf/.../Function}$ parameter.

In addition to specifying the time features, you need to specify the overall time sequence for *Sequential Time Mode*.

**First example**

Here is an example, a Time Feature called *ArmRot* that describes a constant rotation:

```plaintext
s:Tf/ArmRot/Function = "Linear deg"
d:Tf/ArmRot/Rate = 2. deg/ms
d:Tf/ArmRot/StartValue = 0.0 deg
d:Tf/ArmRot/RepetitionInterval = 50. ms
```

TOPAS automatically creates another parameter:

```plaintext
s:Tf/ArmRot/Value
```

and updates this parameter to the appropriate value for a given time.

You can then use this value to affect a component position through a statement such as:
Linear, Sine, Cosine and Sqrt Functions

For **Dimensioned Double** or **Unitless** values, the Function can be any one of:

<table>
<thead>
<tr>
<th>Function</th>
<th>Value(Time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>StartValue + Rate * Time</td>
</tr>
<tr>
<td>Sine</td>
<td>Sine (StartValue + Rate * Time)</td>
</tr>
<tr>
<td>Cosine</td>
<td>Cosine (StartValue + Rate * Time)</td>
</tr>
<tr>
<td>Sqrt</td>
<td>Sqrt (StartValue + Rate * Time)</td>
</tr>
</tbody>
</table>

If the value is Dimensioned Double, you must also provide a unit, such as the `deg` in:

```
s:Tf/ArmRot/Function = "Linear deg"
```

You must provide appropriate **StartValue** and **Rate** parameters, such as:

```
d:Tf/ArmRot/Rate = 2. deg/ms
d:Tf/ArmRot/StartValue = 0.0 deg
```

You must also provide a **RepetitionInterval**, the time interval after which the function will return to the **StartValue**.

**Step Function**

With a Step function, you can control **any** type of parameter value. You define a set of times at which to change value, and a value for each of those times. The first value you provide specifies the starting value (the value at time zero).

Here is an example of a Step time feature that controls a String:

```
s:Tf/ImageName/Function = "Step"
dv:Tf/ImageName/Times = 3 10 20 30 ms
dv:Tf/ImageName/Values = 3 "lung-1" "lung-2" "lung-3"
```

- The first value is used for times 0 to 10 ms.
- The second value is used for times 10 to 20 ms.
- The third value is used for times 20 to 30 ms.
- After 30 ms, the value cycles back to the first value.

Note that whereas continuous functions (Linear, Sine, Cosine and Sqrt) include a **RepetitionInterval**, Step Functions do not. They just cycle back to the first Value after the last of the Times is reached.

Here is an example of a Step time feature that controls a Boolean:

```
s:Tf/ScoringOnOff/Function="Step"
dv:Tf/ScoringOnOff/Times =10 10 20 30 40 50 60 70 80 90 100 ms
dv:Tf/ScoringOnOff/Values=10 "true" "false" "true" "false" "false" "true" "true" "true" "false" "true"
```

Note that:

- **Tf/.../Times** is always of type `dv:` and has unit of time.
- \texttt{Tf/.../Values} is a vector of whatever type the function controls.

Any individual member of the \texttt{Values} parameter vector can itself be a parameter, such as:

\begin{verbatim}
        bv:Tf/ScoringOnOff/Values=4 "true" "false" Some_Other_Boolean_Parameter_Name "false"
\end{verbatim}

### Combining Time Features for Complex Behaviors

You can add or multiply time feature \texttt{Value} parameters just as you can add or multiply any other kind of parameter. For example, here is how the number of histories in a run can be controlled by both a beam current and a beam weight:

```
\begin{verbatim}
        s:Tf/BeamCurrent/Function = "Step"
        dv:Tf/BeamCurrent/Times = 1 10 ms
        iv:Tf/BeamCurrent/Values = 1 10

        s:Tf/BeamWeight/Function = "Step"
        dv:Tf/BeamWeight/Times = 10 1 2 3 4 5 6 7 8 9 10 ms
        iv:Tf/BeamWeight/Values = 10 1 1 1 2 2 2 2 4 4 4

        i:Tf/BCM/Value = Tf/BeamWeight/Value * Tf/BeamCurrent/Value
        i:So/MySource/NumberOfHistoriesInRun = Tf/BCM/Value
\end{verbatim}
```

By combining \texttt{Step} time features with other time features, you can control complex sequences. The following from \texttt{PurgingMagnet_move.txt} moves a box first in one direction and then in the other:

```
\begin{verbatim}
        s:Tf/BackForward/Function = "Step"
        dv:Tf/BackForward/Times = 2 100.0 200.0 ms
        dv:Tf/BackForward/Values = 2 Tf/BackStep/Value Tf/ForwardStep/Value mm

        s:Tf/BackStep/Function = "Linear mm"
        d:Tf/BackStep/Rate = 3 mm/ms
        d:Tf/BackStep/StartValue = 0.0 mm
        d:Tf/BackStep/RepetitionInterval = 100.0 ms

        s:Tf/ForwardStep/Function = "Linear mm"
        d:Tf/ForwardStep/Rate = -3 mm/ms
        d:Tf/ForwardStep/StartValue = 300.0 mm
        d:Tf/ForwardStep/RepetitionInterval = 100.0 ms
\end{verbatim}
```

Some complex examples of time features are in examples/Nozzle. While we have had examples of double scattering and pencil beam scanning for some time, those examples have included proprietary IBA information, so could not be generally shared. The examples found in examples/Nozzle have no vendor confidential information.
Example | Description
--- | ---
*RasterScanningPattern.txt* | Time Features for controlling the dipole magnets are implemented. The time varying magnet will scan rectangle fields in a raster pattern.

*ScanningStationaryTarget.txt* | In addition to *RasterScanningPattern.txt*, a water phantom including a plane target is added.

*ScanningTarget-MovingHorizontal.txt* | The perpendicularly moving target is defined. In order to make protons follow the moving target, compensated Time Features for the dipole magnets are implemented. The execution of this file will show the moving target in horizontal direction and the proton beams tracking the moving target.

*ScanningTarget-MovingInDepth.txt* | To trace the target moving along with the depth, the changes of proton’s incident energy should be synchronized with the motion.

*ScanningNozzle.txt* | All geometry for the scanning nozzle is defined. The nozzle consists of magnet systems, for example, two quadrupole magnets and two dipole magnets in helium gas filled beam pipe and various monitoring chambers. Magnet fields are set to zero in this parameter file.

*ScatteringNozzle.txt* | All geometry for the scattering nozzle is defined.

*ScatteringNozzle_run.txt* | Range Modulator Wheel rotates over time and scatterers move in and out of the beam.

**Warning:** Take care when mixing Phase Space Sources with Time Features. While TOPAS can save the current TOPAS time to a phase space file, this time is not automatically applied when reading particles back in from phase space. Thus, if you want to correct replay source particles that were recorded with time features, it is your responsibility to apply the identical time features during the play back simulation. Some additional notes:

- Do not attempt to change the name of the phase space file over time. Save and replay all particles from a single phase space file.

- Do not use *Random Time Mode*. The randomly generated times during playback will not necessarily match the randomly generated times that were saved to the phase space. Only use *Fixed Time Mode* or *Sequential Time Mode*.

A future version of TOPAS will provide more tools to synchronize and check playback time features.
CHAPTER 17

Variance Reduction

Introduction

Please note that Variance Reduction (VR) is highly dependent on your specific geometry. Approach these features with caution and test all variance reduced setups against an equivalent setup without variance reduction.

You should also review the Geant4 document that describes which cases are problematic here.

To enable the particle split applied to protons:

```
b:Vr/UseVarianceReduction = "true"
b:Vr/ParticleSplit/Active = "true"
sv:Vr/ParticleSplit/ParticleName = 1 "proton"
```

Specify the Split Geometry

The geometry for variance reduction must be in a parallel world. The type of component can be any standard solid (Dividable Components or Generic Components). The geometry must consist of a geometry component with a set of geometry sub-components as daughters. The sub-components must be located in such a way that the boundaries coincide. The split process or Russian roulette will occur at these boundaries. In the next figure a simple scheme is shown.
Time Features can be used to move or rotate the component or sub-components. But there is a restriction: the implementation of VR does not allow you to change the dimensions of the component and sub-components.

To set the geometry for VR:

```
s:Vr/ParticleSplit/Component = "MyComponent"
s:Vr/ParticleSplit/SubComponents = n "MySubComp_1" ... "MySubComp_n"
```

Define the Splitting Technique

There are three variance reduction techniques available:

- **GeometricalParticleSplit**
- **ImportanceSampling**
- **WeightWindow**

To chose a technique, use for example:

```
s:Vr/ParticleSplit/Type = "GeometricalParticleSplit"
```

Geometrical Particle Splitting

TOPAS variance reduction is further described in:


This technique was designed for heavy charged particles. In this implementation, you must specify whether the beam entering into the sub-component has cylindrical symmetry or not. This is because the particles may or may not be randomly redistributed around the SplitAxis.

The Russian roulette is applied in a particular direction. That is, at the split plane and prior to being split, the particle is subject to the Russian roulette if its direction does not point towards a Region of Interest (ROI). Then the radius of the ROI and its position on the SplitAxis must to be defined too. Further, the Russian roulette can be turned on/off at specific surfaces between sub-components.
To set whether the region at each sub-component is symmetric or not and to define the corresponding split number:

In addition for this technique, geometrical Russian roulette will be played if a particle leaves the component or the world in a scheme similar to the Importance Sampling technique.

**Importance Sampling**

In this technique, an importance value is assigned to each sub-component. If a particle is transported into a sub-component with a higher importance, then the particle is split. If it is transported into a sub-component with a lower importance, then Russian roulette is played. By default an importance value of 1 is automatically assigned to the parent component and to the world.

**Warning:** It is desirable for the thickness of the sub-components to be similar to the mean free path of the physical process to be biased.

The sub-component importance values are defined by hand. For example, to split the particles by a factor of 2 between subsequent sub-components, one must to define:
Weight Window

In this technique, the split process or Russian roulette will be applied depending on the statistical weight of the particle. Every time that a particle crosses from a sub-component to the next one, the statistical weight is evaluated:

- Particles with weights greater than a lower bound and smaller than an upper bound will be tracked normally.
- Particles with weights smaller than a lower bound will be subject to Russian roulette. If it survives, the particle is tracked normally but its weight is increased.
- Particles with weights greater than an upper bound will be split, and the new particles will be assigned lower weights.

The split number and Russian roulette criteria are internally calculated from an energy map, a weight map, an upper limit factor and a survival factor. In simple geometries the maps can be input by hand.

The user must provide a double vector with upper energy bounds and a unitless vector with lower weight bounds for every sub-component: WeightMap and EnergyMap. The inverse of a parameter named MaximumSplitNumber (100 by default) is used to specify the minimum survival probability to be used in Russian roulette. The parameter PlaceOfAction states whether the split process (or Russian roulette) will occur at the sub-component boundaries, at physics interactions or at both.

The follow configuration is equivalent to importance sampling with importance generator of 2:

```plaintext
w:Vr/ParticleSplit/Type = "WeightWindow"
w:Vr/ParticleSplit/WeightMap = 4 1. 1. 0.125 0.0615
dv:Vr/ParticleSplit/EnergyMap = 4 1. 1. 1. 1. GeV
u:Vr/ParticleSplit/UpperLimitFactor = 1
u:Vr/ParticleSplit/SurvivalFactor = 1
i:Vr/ParticleSplit/MaximumSplitNumber = 100
s:Vr/ParticleSplit/PlaceOfAction = "onBoundary"
#Others options of PlaceOfAction: "OnCollision" and "OnBoundaryAndCollision"
```
Tracking Only Specific Particles

In this option the particles are eliminated just after they were created. The user can choose which particles will be tracked in all components. Nevertheless, the user can specify if a component is going to be neglected. That is, all particles are tracked in such components. This option can be useful when the contribution of certain particles is negligible to the final scored quantity. But it must be used with caution. This option is not a variance reduction.

To eliminate particles other than protons and electrons in all components but in component named WaterPhantom:

```plaintext
b:Vr/KillOtherParticles/Active = "True"
sv:Vr/KillOtherParticles/HaveNoEffectInComponentsNamed = 1 "WaterPhantom"
sv:Vr/KillOtherParticles/OnlyTrackParticlesNamed = 2 "proton" "e-
```

Secondary Biasing

The split of secondary particles created after an electromagnetic interaction is also supported. A common example is the split of secondary photons created in the bremsstrahlung process for conventional radiotherapy simulations.

This variance reduction works per electromagnetic physical process per region. Physics Regions allow multiple components to have specific production cuts. This is useful in complex geometry setups to improve the computational speed by assigning high production cuts in regions where detailed simulation is unimportant. To assign a region to a component:

```plaintext
e:Ge/MyComponent/AssignToRegionNamed = "MyRegion"
```

The region MyRegion is automatically created if it does not exist. The next step is to set the secondary biasing option:

```plaintext
e:Vr/ParticleSplit/Type = "SecondaryBiasing"
```

Then three vectors must be defined. One with the name of the electromagnetic processes, one with the split number for each process and one with the maximum energies for each processes. The biased particles with energies larger than these values are subject to Russian roulette:

```plaintext
sv:Vr/ParticleSplit/ForRegion/MyRegion/ProcessesNamed = 2 "eBrem" "compt"
uv:Vr/ParticleSplit/ForRegion/MyRegion/SplitNumber = 2 100 10
dv:Vr/ParticleSplit/ForRegion/MyRegion/MaximumEnergy = 2 6.0 0.511 MeV
```
If suitable, further CPU time can be saved with a directional Russian roulette for secondary particles created with split (analogous to *Geometrical Particle Splitting*). For that, a reference component must be chosen:

```
@Vr/ParticleSplit/ReferenceComponent = "Target"
```

And the directional filter is applied:

```
@Vr/ParticleSplit/ForRegion/MyRegion/DirectionalSplitLimits = 2 1.0 1.0 m
@Vr/ParticleSplit/ForRegion/MyRegion/DirectionalSplitRadius = 2 5.0 5.0 cm
```

Figure: Biasing particle of secondary photons after a bremsstrahlung process. On the left, no directional Russian roulette is applied. On the right, a directional Russian roulette is applied.
# Demonstrates all allowed parameter forms
# and dumps results to TsDumpParameters.html

b:Ts/DumpNonDefaultParameters = "True"
sv:Ph/Default/Modules = 1 "Transportation_Only"

# Parameters used in expressions below

d:A_Double1 = 100. cm
d:A_Double5 = 500. cm
d:A_Double10 = 1000. cm
d:A_Double20 = 2000. cm
w:A_Unitless1 = 1.
w:A_Unitless5 = 5.
w:A_Unitless10 = 10.
w:A_Unitless20 = 20.
i:A_Integer1 = 1
i:A_Integer5 = 5
i:A_Integer10 = 10
i:A_Integer20 = 20
s:A_StringTen = "Ten"
b:A_BooleanTrue = "True"
dv:A_DoubleVector1 = 3 1. 1. 1. m
uv:A_UnitlessVector1 = 3 1. 1. 1.
iv:A_IntegerVector1 = 3 1 1 1
sv:A_StringVector1 = 3 "One" "One" "One"

# Double parameters: correct answer is always 10m

d:A_DoubleFromValue = 10. m
d:A_DoubleFromValuePlusDouble = 0. m + A_Double10
d:A_DoubleFromValueMinusDouble = 20. m - A_Double10
A DOUBLE FROM VALUE TIMES UNITLESS = 10. m * A_UNITLESS1
A DOUBLE FROM VALUE TIMES INTEGER = 10. m * A_INTEGER1
A DOUBLE FROM VALUE TIMES DOUBLE = 1. * A_DOUBLE10 m
A DOUBLE FROM DOUBLE = A_DOUBLE10 m
A DOUBLE FROM DOUBLE TIMES VALUE = A_DOUBLE10 m * 1.
A DOUBLE FROM DOUBLE TIMES UNITLESS = A_DOUBLE10 m * A_UNITLESS1
A DOUBLE FROM DOUBLE TIMES INTEGER = A_DOUBLE10 m * A_INTEGER1
A DOUBLE FROM UNITLESS TIMES VALUE = A_UNITLESS1 * 10. m
A DOUBLE FROM INTEGER TIMES VALUE = A_INTEGER1 * 10. m
A DOUBLE FROM DOUBLE PLUS VALUE = A_DOUBLE10 + 0. m
A DOUBLE FROM DOUBLE MINUS VALUE = A_DOUBLE20 - 10. m
A DOUBLE FROM DOUBLE PLUS DOUBLE = A_DOUBLE5 + A_DOUBLE5 m
A DOUBLE FROM DOUBLE MINUS DOUBLE = A_DOUBLE20 - A_DOUBLE10 m

# Unitless parameters: correct answer is always 10
A UNITLESS FROM VALUE = 10.
A UNITLESS FROM VALUE PLUS UNITLESS = 0. + A_UNITLESS10
A UNITLESS FROM VALUE MINUS UNITLESS = 20. - A_UNITLESS10
A UNITLESS FROM VALUE TIMES UNITLESS = 1. * A_UNITLESS10
A UNITLESS FROM VALUE PLUS INTEGER = 0. + A_INTEGER10
A UNITLESS FROM VALUE MINUS INTEGER = 20. - A_INTEGER10
A UNITLESS FROM VALUE TIMES INTEGER = A_UNITLESS10 * 1.
A UNITLESS FROM INTEGER TIMES VALUE = A_INTEGER10 * 1.
A UNITLESS FROM UNITLESS PLUS UNITLESS = A_UNITLESS5 + A_UNITLESS5
A UNITLESS FROM UNITLESS MINUS UNITLESS = A_UNITLESS20 - A_UNITLESS10
A UNITLESS FROM UNITLESS TIMES UNITLESS = A_UNITLESS10 * A_UNITLESS1
A UNITLESS FROM UNITLESS PLUS INTEGER = A_UNITLESS5 + A_INTEGER5
A UNITLESS FROM UNITLESS MINUS INTEGER = A_UNITLESS20 - A_INTEGER10
A UNITLESS FROM UNITLESS TIMES INTEGER = A_UNITLESS10 * A_INTEGER1
A UNITLESS FROM INTEGER PLUS UNITLESS = A_INTEGER5 + A_UNITLESS5
A UNITLESS FROM INTEGER MINUS UNITLESS = A_INTEGER20 - A_UNITLESS10
A UNITLESS FROM INTEGER TIMES UNITLESS = A_INTEGER10 * A_UNITLESS1
A UNITLESS FROM INTEGER PLUS INTEGER = A_INTEGER5 + A_INTEGER5
A UNITLESS FROM INTEGER MINUS INTEGER = A_INTEGER20 - A_INTEGER10
A UNITLESS FROM INTEGER TIMES INTEGER = A_INTEGER10 * A_INTEGER1

# Integer parameters: correct answer is always 10
A_INTEGER FROM VALUE = 10
A_INTEGER FROM VALUE PLUS INTEGER = 0 + A_INTEGER10
A INTEGER FROM VALUE MINUS INTEGER = 20 - A_INTEGER10
A INTEGER FROM VALUE TIMES INTEGER = 1 * A_INTEGER10
A INTEGER FROM INTEGER = A_INTEGER10
A INTEGER FROM INTEGER PLUS VALUE = A_INTEGER10 + 0
A INTEGER FROM INTEGER MINUS VALUE = A_INTEGER20 - 10
A INTEGER FROM INTEGER TIMES VALUE = A_INTEGER10 * 1
A INTEGER FROM INTEGER PLUS INTEGER = A_INTEGER5 + A_INTEGER5
A INTEGER FROM INTEGER MINUS INTEGER = A_INTEGER20 - A_INTEGER10
A INTEGER FROM INTEGER TIMES INTEGER = A_INTEGER10 * A_INTEGER1

# Boolean parameters: correct answer is always 1 (= True)
A_BOOLEAN FROM VALUE = "True"
b: A_BooleanFromBoolean = A_BooleanFromValue
b: A_BooleanFromBooleanTimesBoolean = A_BooleanFromValue * A_BooleanTrue

# String parameters: correct answer is always Ten, 10
# or some combination of the two such as TenTen, Ten10, 1010, etc.
s: A_StringFromValue = "Ten"
s: A_StringFromValuePlusInteger = "Ten" + A_Integer10
s: A_StringFromValuePlusString = "Ten" + A_StringTen
s: A_StringFromInteger = A_Integer10
s: A_StringFromString = A_StringTen
s: A_StringFromIntegerPlusValue = A_Integer10 + "Ten"
s: A_StringFromValuePlusUnitlessVector = A_StringTen + "Ten"
s: A_StringFromIntegerPlusInteger = A_Integer10 + A_Integer10
s: A_StringFromStringPlusInteger = A_StringTen + A_Integer10
s: A_StringFromStringPlusString = A_StringTen + A_StringTen

# Double Vectors: correct answer is always 10. 5. 1. m
dv: A_DoubleVectorFromValue = 3 10. 5. 1. m
dv: A_DoubleVectorFromValuePlusDouble = 3 9. 4. 0. m + A_Double1
dv: A_DoubleVectorFromValueMinusDouble = 3 11. 6. 2. m - A_Double1
dv: A_DoubleVectorFromValueTimesUnitless = 3 10. 5. 1. m * A_Unitless1
dv: A_DoubleVectorFromValueTimesInteger = 3 10. 5. 1. m * A_Integer1
dv: A_DoubleVectorFromValueTimesUnitlessVector = 3 10. 5. 1. m * A_UnitlessVector1
dv: A_DoubleVectorFromValueTimesIntegerVector = 3 10. 5. 1. m * A_IntegerVector1
dv: A_DoubleVectorFromValueTimesDouble = 3 10. 5. 1. * A_Double1 m
dv: A_DoubleVectorFromValueTimesDoubleVector = 3 10. 5. 1. * A_DoubleVector1 m
dv: A_DoubleVectorFromScaleTimesDoubleVector = 1. * A_DoubleVectorFromValue m
dv: A_DoubleVectorFromIntegerTimesDoubleVector = A_INTEGERFromValue * A_˓
  DoubleVectorFromValue m
dv: A_DoubleVectorFromUnitlessTimesDoubleVector = A_UnitlessFromValue * A_˓
  DoubleVectorFromValue m

# Unitless Vectors: correct answer is always 10 5 1
uv: A_UnitlessVectorFromValue = 3 10. 5. 1.
uv: A_UnitlessVectorFromValuePlusUnitless = 3 9. 4. 0. + A_Unitless1
uv: A_UnitlessVectorFromValueMinusUnitless = 3 11. 6. 2. - A_Unitless1
uv: A_UnitlessVectorFromValueTimesUnitless = 3 10. 5. 1. * A_Unitless1
uv: A_UnitlessVectorFromValueMinusInteger = 3 11. 6. 2. - A_Integer1
uv: A_UnitlessVectorFromValueTimesInteger = 3 10. 5. 1. * A_Integer1
uv: A_UnitlessVectorFromValuePlusUnitlessVector = 3 9. 4. 0. + A_UnitlessVector1
uv: A_UnitlessVectorFromValueMinusUnitlessVector = 3 11. 6. 2. - A_UnitlessVector1
uv: A_UnitlessVectorFromValueTimesUnitlessVector = 3 10. 5. 1. * A_UnitlessVector1
uv: A_UnitlessVectorFromValueMinusIntegerVector = 3 11. 6. 2. - A_IntegerVector1
uv: A_UnitlessVectorFromValueTimesIntegerVector = 3 10. 5. 1. * A_IntegerVector1
uv: A_UnitlessVectorFromUnitlessVector = A_UnitlessVectorFromValue
uv: A_UnitlessVectorFromScaleTimesUnitlessVector = 1. * A_UnitlessVectorFromValue
uv: A_UnitlessVectorFromUnitlessTimesUnitlessVector = A_UnitlessFromValue * A_˓
  UnitlessVectorFromValue
uv: A_UnitlessVectorFromIntegerTimesUnitlessVector = A_INTEGERFromValue * A_˓
  UnitlessVectorFromValue

# Integer Vectors: correct answer is always 10 5 1
### A_IntegerVectorFromValue

\[
\text{A_IntegerVectorFromValue} = 3 \ 10 \ 5 \ 1
\]

### A_IntegerVectorFromValuePlusInteger

\[
\text{A_IntegerVectorFromValuePlusInteger} = 3 \ 9 \ 4 \ 0 + \text{A_Integer1}
\]

### A_IntegerVectorFromValueMinusInteger

\[
\text{A_IntegerVectorFromValueMinusInteger} = 3 \ 11 \ 6 \ 2 - \text{A_Integer1}
\]

### A_IntegerVectorFromValueTimesInteger

\[
\text{A_IntegerVectorFromValueTimesInteger} = 3 \ 10 \ 5 \ 1 \times \text{A_Integer1}
\]

### A_IntegerVectorFromValuePlusIntegerVector

\[
\text{A_IntegerVectorFromValuePlusIntegerVector} = 3 \ 9 \ 4 \ 0 + \text{A_IntegerVector1}
\]

### A_IntegerVectorFromValueMinusIntegerVector

\[
\text{A_IntegerVectorFromValueMinusIntegerVector} = 3 \ 11 \ 6 \ 2 - \text{A_IntegerVector1}
\]

### A_IntegerVectorFromValueTimesIntegerVector

\[
\text{A_IntegerVectorFromValueTimesIntegerVector} = 3 \ 10 \ 5 \ 1 \times \text{A_IntegerVector1}
\]

### A_IntegerVectorFromIntegerVector

\[
\text{A_IntegerVectorFromIntegerVector} = \text{A_IntegerVectorFromValue}
\]

### A_IntegerVectorFromScaleTimesIntegerVector

\[
\text{A_IntegerVectorFromScaleTimesIntegerVector} = 1 \times \text{A_IntegerVectorFromValue}
\]

### A_IntegerVectorFromIntegerTimesIntegerVector

\[
\text{A_IntegerVectorFromIntegerTimesIntegerVector} = \text{A_IntegerFromValue} \times \text{A_IntegerVectorFromValue}
\]

### Boolean Vectors: correct answer is always 1 0 1 (= True False True)

\[
\text{A_BooleanVectorFromValue} = 3 \ \text{"True"} \ \text{"False"} \ \text{"True"}
\]

\[
\text{A_BooleanVectorFromBooleanVector} = \text{A_BooleanVectorFromValue}
\]

### String Vectors: correct answers vary by case

\[
\text{A_StringVectorFromValue} = 3 \ \text{"Ten"} \ \text{"Five"} \ \text{"One"}
\]

\[
\text{A_StringVectorFromValuePlusInteger} = 3 \ \text{"Ten"} \ \text{"Five"} \ \text{"One"} + \text{A_Integer10}
\]

\[
\text{A_StringVectorFromValuePlusString} = 3 \ \text{"Ten"} \ \text{"Five"} \ \text{"One"} + \text{A_StringTen}
\]

\[
\text{A_StringVectorFromStringVector} = \text{A_StringVectorFromValue}
\]

\[
\text{A_StringVectorFromValuePlusIntegerVector} = 3 \ \text{"Ten"} \ \text{"Five"} \ \text{"One"} + \text{A_IntegerVector1}
\]

\[
\text{A_StringVectorFromValuePlusStringVector} = 3 \ \text{"Ten"} \ \text{"Five"} \ \text{"One"} + \text{A_StringVector1}
\]

---

**DividedComponents.txt**

### Demonstrates use of the Divided Components, TsBox, TsCylinder and TsSphere.

\[
\text{d:Ge/World/HLX} = 2.0 \text{ m}
\]

\[
\text{d:Ge/World/HLY} = 2.0 \text{ m}
\]

\[
\text{d:Ge/World/HLZ} = 2.0 \text{ m}
\]

\[
\text{b:Ge/World/Invisible} = \text{"True"}
\]

### Save time by turning off overlap check

\[
\text{b:Ge/CheckForOverlaps} = \text{"False"}
\]

\[
\text{s:Ge/TestBox/Parent} = \text{"World"}
\]

\[
\text{s:Ge/TestBox/Type} = \text{"TsBox"}
\]

\[
\text{s:Ge/TestBox/Material} = \text{"G4_WATER"}
\]

\[
\text{d:Ge/TestBox/HLX} = 7.0 \text{ cm}
\]

\[
\text{d:Ge/TestBox/HLY} = 7.0 \text{ cm}
\]

\[
\text{d:Ge/TestBox/HLZ} = 9.0 \text{ cm}
\]

\[
\text{d:Ge/TestBox/TransX} = 0. \text{ cm}
\]

\[
\text{d:Ge/TestBox/TransY} = 0. \text{ cm}
\]

\[
\text{d:Ge/TestBox/TransZ} = 0. \text{ cm}
\]

\[
\text{d:Ge/TestBox/RotX} = 0. \text{ deg}
\]

\[
\text{d:Ge/TestBox/RotY} = 0. \text{ deg}
\]

\[
\text{d:Ge/TestBox/RotZ} = 0. \text{ deg}
\]

\[
\text{s:Ge/TestBox/Color} = \text{"blue"}
\]

\[
\text{i:Ge/TestBox/XBins} = 3
\]

\[
\text{i:Ge/TestBox/YBins} = 4
\]

\[
\text{i:Ge/TestBox/ZBins} = 5
\]

\[
\text{s:Ge/TestCylinder/Parent} = \text{"World"}
\]

---

146 Chapter 18. Basic
<table>
<thead>
<tr>
<th>Ge/TestCylinder/Type</th>
<th>&quot;TsCylinder&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/TestCylinder/Material</td>
<td>&quot;G4_WATER&quot;</td>
</tr>
<tr>
<td>Ge/TestCylinder/RMin</td>
<td>0.0 cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/RMax</td>
<td>8.0 cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/HL</td>
<td>10.0 cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/SPhi</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/DPhi</td>
<td>360. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransX</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransY</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransZ</td>
<td>25. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotX</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotY</td>
<td>0.0 deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotZ</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/Color</td>
<td>&quot;blue&quot;</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransX</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransY</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/TransZ</td>
<td>25. cm</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotX</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotY</td>
<td>0.0 deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/RotZ</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestCylinder/Color</td>
<td>&quot;blue&quot;</td>
</tr>
<tr>
<td>Ge/TestCylinder/RBins</td>
<td>2</td>
</tr>
<tr>
<td>Ge/TestCylinder/PhiBins</td>
<td>8</td>
</tr>
<tr>
<td>Ge/TestCylinder/ZBins</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ge/TestSphere/Parent</th>
<th>&quot;World&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/TestSphere/Type</td>
<td>&quot;TsSphere&quot;</td>
</tr>
<tr>
<td>Ge/TestSphere/Material</td>
<td>&quot;G4_WATER&quot;</td>
</tr>
<tr>
<td>Ge/TestSphere/RMin</td>
<td>0.0 cm</td>
</tr>
<tr>
<td>Ge/TestSphere/RMax</td>
<td>10.0 cm</td>
</tr>
<tr>
<td>Ge/TestSphere/SPhi</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/DPhi</td>
<td>360. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/STheta</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/DTheta</td>
<td>180. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/TransX</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestSphere/TransY</td>
<td>0. cm</td>
</tr>
<tr>
<td>Ge/TestSphere/TransZ</td>
<td>50. cm</td>
</tr>
<tr>
<td>Ge/TestSphere/RotX</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/RotY</td>
<td>0.0 deg</td>
</tr>
<tr>
<td>Ge/TestSphere/RotZ</td>
<td>0. deg</td>
</tr>
<tr>
<td>Ge/TestSphere/Color</td>
<td>&quot;blue&quot;</td>
</tr>
<tr>
<td>Ge/TestSphere/RBins</td>
<td>3</td>
</tr>
<tr>
<td>Ge/TestSphere/PhiBins</td>
<td>4</td>
</tr>
<tr>
<td>Ge/TestSphere/ThetaBins</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gr/ViewA/Type</th>
<th>&quot;OpenGL&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gr/ViewA/WindowSizeX</td>
<td>1024</td>
</tr>
<tr>
<td>Gr/ViewA/WindowSizeY</td>
<td>768</td>
</tr>
<tr>
<td>Gr/ViewA/Theta</td>
<td>55 deg</td>
</tr>
<tr>
<td>Gr/ViewA/Phi</td>
<td>20 deg</td>
</tr>
<tr>
<td>Gr/ViewA/Projection</td>
<td>&quot;Perspective&quot;</td>
</tr>
<tr>
<td>Gr/ViewA/PerspectiveAngle</td>
<td>30 deg</td>
</tr>
<tr>
<td>Gr/ViewA/Zoom</td>
<td>1.8</td>
</tr>
<tr>
<td>Gr/ViewA/HiddenLineRemovalForTrajectories</td>
<td>&quot;True&quot;</td>
</tr>
</tbody>
</table>

Ph/Default/Modules = 1 "g4em-standard_opt0"

Ts/PauseBeforeQuit = "True"
Emittance_Gaussian.txt

Ge/World/Material = "Vacuum"

#--- Beam
Ge/Example/Component = "BeamPosition"
Ge/Example/BeamParticle = "proton"
Ge/Example/BeamEnergy = 94.00 MeV
Ge/Example/BeamEnergySpread = 1.0
Ge/Example/NumberOfHistoriesInRun = 10000

#----- Primary: Emittance beam
Ge/Example/Type = "Emittance"
Ge/Example/Distribution = "BiGaussian"
Ge/Example/SigmaX = 2.0 mm
Ge/Example/SigmaXprime = 0.0032 #32 mrad
Ge/Example/CorrelationX = -0.8
Ge/Example/SigmaY = 2.0 mm
Ge/Example/SigmaYPrime = 0.0032
Ge/Example/CorrelationY = 0.4

# Beam position (S)
Ge/BeamPosition/Parent = "World"
Ge/BeamPosition/Type = "Group"
Ge/BeamPosition/TransX = 0. m
Ge/BeamPosition/TransY = 0. m
Ge/BeamPosition/TransZ = 3.001 m
#flipped cause the beam flies 0 to +z.
Ge/BeamPosition/RotX = 180. deg
Ge/BeamPosition/RotY = 0. deg
Ge/BeamPosition/RotZ = 0. deg

#---- Verbosity
Ts/TrackingVerbosity = 0

#---- Sequence
Ge/CheckForUnusedComponents = "False"
Ts/ShowCPUTime = "true"
Ts/ShowHistoryCountAtInterval = 1000
Ts/Seed = 10

#--- Physics --
Ph/Default/Modules = 1 "g4em-standard_opt0"

#---- Constants
Ge/MyBeamSizeInX = 20.0 cm
Ge/MyBeamSizeInY = 20.0 cm

#--- Beam Box
Ge/PlaneAtBeamPosition/Type = "TsBox"
Ge/PlaneAtBeamPosition/Material = "Vacuum"
Ge/PlaneAtBeamPosition/Parent = "World"
Ge/PlaneAtBeamPosition/HLX = 0.5 cm + Ge/MyBeamSizeInX
Ge/PlaneAtBeamPosition/HLY = 0.5 cm + Ge/MyBeamSizeInY
Ge/PlaneAtBeamPosition/HLZ = 0.05 mm
Emittance_Twiss.txt

```plaintext
includeFile = Emittance_Gaussian.txt

### You can choose one from these three options
s:So/Example/Distribution = "twiss_gaussian"
#s:So/Example/Distribution = "twiss_kv"
#s:So/Example/Distribution = "twiss_waterbag"

u:So/Example/AlphaX = 0.2
d:So/Example/BetaX = 600.0 mm #
d:So/Example/EmittanceX = 0.01 mm #we don't multiply pi

u:So/Example/AlphaY = 2.5
d:So/Example/BetaY = 1400.0 mm
d:So/Example/EmittanceY = 0.02 mm

### twiss gaussian option only
#0.9 means that 90% particles will be included in ellipse circle
u:So/Example/ParticleFractionX = 0.90
u:So/Example/ParticleFractionY = 0.90

s:Sc/Beam/OutputFile = "BeamTwiss"
```

FlatteningFilter.txt

```plaintext
# This is not any particular actual flattening filter.
# It just demonstrates the general principle of how to
# combine two different kinds of polycones to create
# a shape that no single polycone can represent.

b:Ge/World/Invisible = "True"
```
Isotope.txt

# Example of creating an element from specific abundances

```
sv:Ma/MyUranium/Components = 1 "MyElU"
sv:Ma/MyUranium/Fractions = 1 1.0
d:Ma/MyUranium/Density = 18.95 g/cm3
sv:Ma/MyUranium/DefaultColor = "green"

sv:El/MyElU/Symbol = "MyElU"
sv:El/MyElU/IsotopeNames = 2 "U235" "U238"
sv:El/MyElU/IsotopeAbundances = 2 90. 10.
```

```
i:Is/U235/Z = 92
d:Is/U235/N = 235
d:Is/U235/A = 235.01 g/mole
id:Is/U238/Z = 92
d:Is/U238/N = 238
d:Is/U238/A = 238.03 g/mole
```

```
s:Ge/MyBox/Type = "TsBox"
s:Ge/MyBox/Material = "MyUranium"
s:Ge/MyBox/Parent = "World"
s:Ge/MyBox/HLX = 2.5 m
d:Ge/MyBox/HLY = 2. m
```
LayeredMassGeometry.txt

# Demonstrates use of Layered Mass Geometry, 
# a Geant4 feature whereby there can be material in a parallel world.

s:Ge/World/Material = "Vacuum" 
d:Ge/World/HLX = 2.0 m 
d:Ge/World/HLY = 2.0 m 
d:Ge/World/HLZ = 2.0 m 
b:Ge/World/Invisible = "TRUE"

sv:Ph/Default/LayeredMassGeometryWorlds = 2 "WaterBox" "IronBox"

s:Ge/AirBox/Parent = "World" 
s:Ge/AirBox/Type = "TsBox" 
s:Ge/AirBox/Material = "Air" 
d:Ge/AirBox/HLX = 1. m 
d:Ge/AirBox/HLY = 1. m
\[
\begin{align*}
\text{d:Ge/AirBox/HLZ} & = 1. \text{ m} \\
\text{d:Ge/AirBox/TransX} & = 0. \text{ m} \\
\text{d:Ge/AirBox/TransY} & = 0. \text{ m} \\
\text{d:Ge/AirBox/TransZ} & = 0. \text{ m} \\
\text{d:Ge/AirBox/RotX} & = 0. \text{ deg} \\
\text{d:Ge/AirBox/RotY} & = 0. \text{ deg} \\
\text{d:Ge/AirBox/RotZ} & = 0. \text{ deg} \\
\text{s:Ge/WaterBox/Parent} & = "World" \\
\text{s:Ge/WaterBox/Type} & = "TsBox" \\
\text{s:Ge/WaterBox/Material} & = "G4\_WATER" \\
\text{d:Ge/WaterBox/HLX} & = .6 \text{ m} \\
\text{d:Ge/WaterBox/HLY} & = .6 \text{ m} \\
\text{d:Ge/WaterBox/HLZ} & = .6 \text{ m} \\
\text{d:Ge/WaterBox/TransX} & = 0. \text{ m} \\
\text{d:Ge/WaterBox/TransY} & = 0. \text{ m} \\
\text{d:Ge/WaterBox/TransZ} & = 0. \text{ m} \\
\text{d:Ge/WaterBox/RotX} & = 0. \text{ deg} \\
\text{d:Ge/WaterBox/RotY} & = 0. \text{ deg} \\
\text{d:Ge/WaterBox/RotZ} & = 0. \text{ deg} \\
\text{b:Ge/WaterBox/isParallel} & = "True" \\
\text{s:Ge/IronBox/Parent} & = "World" \\
\text{s:Ge/IronBox/Type} & = "TsBox" \\
\text{s:Ge/IronBox/Material} & = "Iron" \\
\text{d:Ge/IronBox/HLX} & = .4 \text{ m} \\
\text{d:Ge/IronBox/HLY} & = .4 \text{ m} \\
\text{d:Ge/IronBox/HLZ} & = .4 \text{ m} \\
\text{d:Ge/IronBox/TransX} & = 0. \text{ m} \\
\text{d:Ge/IronBox/TransY} & = 0. \text{ m} \\
\text{d:Ge/IronBox/TransZ} & = 0. \text{ m} \\
\text{d:Ge/IronBox/RotX} & = 0. \text{ deg} \\
\text{d:Ge/IronBox/RotY} & = 0. \text{ deg} \\
\text{d:Ge/IronBox/RotZ} & = 0. \text{ deg} \\
\text{b:Ge/IronBox/isParallel} & = "True" \\
\text{b:Ts/PauseBeforeQuit} & = "True" \\
\text{s:So/Example/Type} & = "Beam" \\
\text{s:So/Example/Component} & = "BeamPosition" \\
\text{s:So/Example/BeamParticle} & = "proton" \\
\text{d:So/Example/BeamEnergy} & = 270. \text{ MeV} \\
\text{u:So/Example/BeamEnergySpread} & = 0.757504 \\
\text{s:So/Example/BeamPositionDistribution} & = "Flat" \\
\text{s:So/Example/BeamPositionCutoffShape} & = "Ellipse" \\
\text{d:So/Example/BeamPositionCutoffX} & = 5 \text{ cm} \\
\text{d:So/Example/BeamPositionCutoffY} & = 20 \text{ cm} \\
\text{s:So/Example/BeamAngularDistribution} & = "None" \\
\text{i:So/Example/NumberOfHistoriesInRun} & = 10 \\
\text{sv:Ph/Default/Modules} & = 1 "g4em-standard_opt0" \\
\text{s:Gr/ViewA/Type} & = "OpenGL" \\
\text{sv:Gr/ViewA/VisibleWorlds} & = 1 "All" \\
\text{i:Gr/ViewA/WindowSizeX} & = 900 \\
\text{i:Gr/ViewA/WindowSizeY} & = 900 \\
\text{d:Gr/ViewA/Theta} & = 90 \text{ deg} \\
\text{d:Gr/ViewA/Phi} & = 0 \text{ deg} 
\end{align*}
\]
OneBox.txt

# Simplest TOPAS example.
# A box in a beam with EM physics.

s:Ge/MyBox/Type = "TsBox"
s:Ge/MyBox/Material = "Air"
s:Ge/MyBox/Parent = "World"
d:Ge/MyBox/HLX = 2.5 m
d:Ge/MyBox/HLY = 2. m
d:Ge/MyBox/HLZ = 1. m
d:Ge/MyBox/TransX = 2. m
d:Ge/MyBox/TransY = 0. m
d:Ge/MyBox/TransZ = 0. m
d:Ge/MyBox/RotX = 0. deg
d:Ge/MyBox/RotY = 0. deg
d:Ge/MyBox/RotZ = 0. deg
sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
b:Gr/ViewA/IncludeAxes = "True"
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 2.

b:Ts/PauseBeforeQuit = "True"

OneBoxRotate.txt

# Demonstrates use of includeFile.
# Overrides the RotX value from OneBox.txt

includeFile = OneBox.txt
d:Ge/MyBox/RotX = 45. deg

OneBoxTranslate.txt

# Demonstrates use of includeFile.
# Translates box by a value of -0.5 times the value
# of TransX that was inherited from the includeFile.

includeFile = OneBox.txt
d:Ge/MyBox/TransX = inheritedValue m * -0.5
PhysicsSetting.txt

# Demonstrates having several physics lists defined.
# While only one physics list can be used in a given TOPAS session,
# you are free to define several different sets in your parameter files.
# The parameter Ph/ListName tells which of these lists is actually used.

s:Ph/ListName = "MyList1"

s:Ph/MyList1/Type = "Geant4_Modular"
s:Ph/MyList1/Modules = 3 "g4decay" "g4em-standard_opt3" "g4h-elastic"
d:Ph/MyList1/CutForGamma = 0.04 mm
d:Ph/MyList1/CutForElectron = 0.03 mm
d:Ph/MyList1/CutForPositron = 0.02 mm
d:Ph/MyList1/CutForProton = 0.01 mm

s:Ph/MyList2/Type = "Geant4_Modular"
s:Ph/MyList2/Modules = 1 "g4em-standard_opt0"
d:Ph/MyList2/CutForGamma = 0.04 mm
d:Ph/MyList2/CutForElectron = 0.03 mm
d:Ph/MyList2/CutForPositron = 0.02 mm
d:Ph/MyList2/CutForProton = 0.01 mm

s:Ph/MyList3/Type = "Geant4_Modular"
s:Ph/MyList3/Modules = 1 "Transportation_Only"
d:Ph/MyList3/CutForGamma = 0.04 mm
d:Ph/MyList3/CutForElectron = 0.03 mm
d:Ph/MyList3/CutForPositron = 0.02 mm
d:Ph/MyList3/CutForProton = 0.01 mm

s:Ph/MyList4/Type = "QGSP_BERT_HP"
d:Ph/MyList4/CutForAllParticles = 0.03 mm
i:Ph/MyList4/Verbosity = 2

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "G4_WATER"
d:Ge/Phantom/HLX = 80.0 cm
d:Ge/Phantom/HLY = 80.0 cm
d:Ge/Phantom/HLZ = 20.0 cm
d:Ge/Phantom/TransX = 0. cm
d:Ge/Phantom/TransY = 0. cm
d:Ge/Phantom/TransZ = -30. cm
d:Ge/Phantom/RotX = 0. deg
d:Ge/Phantom/RotY = 0.0 deg
d:Ge/Phantom/RotZ = 0. deg
s:Ge/Phantom/Color = "blue"
i:Ge/Phantom/XBins = 2

s:Sc/EnergyDepAtPhantom/Quantity = "DoseToWater"
s:Sc/EnergyDepAtPhantom/Component = "Phantom"
b:Sc/EnergyDepAtPhantom/OutputToConsole = "TRUE"
s:Sc/EnergyDepAtPhantom/IfOutputFileAlreadyExists = "Overwrite"

s:Gr/ViewA/Type = "OpenGL"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
ShapeTestWithAllParameters.txt

# Demonstrates all of the standard Geant4 solids.
# Shows all parameters, including optional ones
# (the ones such as TransX or RMin that have default values).
# Details are given in the TOPAS User Guide
# and in chapter 4 of the Geant4 Application Developer's Guide

# Box
s:Ge/DemoBox/Type = "TsBox"
s:Ge/DemoBox/Parent = "World"
s:Ge/DemoBox/Material = "Air"
d:Ge/DemoBox/TransX = 0 cm
d:Ge/DemoBox/TransY = 0 cm
d:Ge/DemoBox/TransZ = 0 cm
d:Ge/DemoBox/RotX = 0 deg
d:Ge/DemoBox/RotY = 0 deg
d:Ge/DemoBox/RotZ = 0 deg
d:Ge/DemoBox/HLX = 30. mm
d:Ge/DemoBox/HLY = 40. mm
d:Ge/DemoBox/HLZ = 60. mm
s:Ge/DemoBox/Color = "white"

# Cylinder
s:Ge/DemoCylinder/Type = "TsCylinder"
s:Ge/DemoCylinder/Parent = "World"
s:Ge/DemoCylinder/Material = "Air"
d:Ge/DemoCylinder/TransX = 0 cm
d:Ge/DemoCylinder/TransY = 0 cm
d:Ge/DemoCylinder/TransZ = 12 cm
d:Ge/DemoCylinder/RotX = 0 deg
d:Ge/DemoCylinder/RotY = 0 deg
d:Ge/DemoCylinder/RotZ = 0 deg
d:Ge/DemoCylinder/RMin = 10 mm
d:Ge/DemoCylinder/RMax = 15 mm
# CutTubs

```plaintext
d:Ge/DemoCutTubs/Type = "G4CutTubs"
d:Ge/DemoCutTubs/Parent = "World"
d:Ge/DemoCutTubs/Material = "Air"
d:Ge/DemoCutTubs/TransX = 0 cm
d:Ge/DemoCutTubs/TransY = 0 cm
d:Ge/DemoCutTubs/TransZ = 24 cm
d:Ge/DemoCutTubs/RotX = 0 deg
d:Ge/DemoCutTubs/RotY = 0 deg
d:Ge/DemoCutTubs/RotZ = 0 deg
d:Ge/DemoCutTubs/RMin = 12 mm
d:Ge/DemoCutTubs/RMax = 20 mm
d:Ge/DemoCutTubs/HL = 30 mm
d:Ge/DemoCutTubs/SPhi = 0 rad
d:Ge/DemoCutTubs/DPhi = 1.5 rad
uv:Ge/DemoCutTubs/LowNorm = 3 0. -0.7 -0.71
uv:Ge/DemoCutTubs/HighNorm = 3 0.7 0. 0.71
d:Ge/DemoCutTubs/Color = "skyblue"
d:Ge/DemoCutTubs/DrawingStyle = "FullWireFrame"
```

# Cons

```plaintext
d:Ge/DemoCons/Type = "G4Cons"
d:Ge/DemoCons/Parent = "World"
d:Ge/DemoCons/Material = "Air"
d:Ge/DemoCons/TransX = 0 cm
d:Ge/DemoCons/TransY = 0 cm
d:Ge/DemoCons/TransZ = 36 cm
d:Ge/DemoCons/RotX = 0 deg
d:Ge/DemoCons/RotY = 0 deg
d:Ge/DemoCons/RotZ = 0 deg
d:Ge/DemoCons/RMin1 = 5 mm
d:Ge/DemoCons/RMax1 = 10 mm
d:Ge/DemoCons/RMin2 = 20 mm
d:Ge/DemoCons/RMax2 = 25 mm
d:Ge/DemoCons/HL = 40 mm
d:Ge/DemoCons/SPhi = 0 rad
d:Ge/DemoCons/DPhi = 1.34 rad
d:Ge/DemoCons/Color = "red"
d:Ge/DemoCons/DrawingStyle = "FullWireFrame"
```

# Para

```plaintext
d:Ge/DemoPara/Type = "G4Para"
d:Ge/DemoPara/Parent = "World"
d:Ge/DemoPara/Material = "Air"
d:Ge/DemoPara/TransX = 0 cm
d:Ge/DemoPara/TransY = 0 cm
d:Ge/DemoPara/TransZ = 48 cm
d:Ge/DemoPara/RotX = 0 deg
d:Ge/DemoPara/RotY = 0 deg
d:Ge/DemoPara/RotZ = 0 deg
d:Ge/DemoPara/HLX = 30. mm
d:Ge/DemoPara/HLY = 40. mm
```
d:Ge/DemoPara/HLZ = 60. mm
d:Ge/DemoPara/Alpha = 0.3 rad
d:Ge/DemoPara/Theta = 0 rad
d:Ge/DemoPara/Phi = 0 rad
s:Ge/DemoPara/Color = "magenta"

# Trd
s:Ge/DemoTrd/Type = "G4Trd"
s:Ge/DemoTrd/Parent = "World"
s:Ge/DemoTrd/Material = "Air"
d:Ge/DemoTrd/TransX = 0 cm
d:Ge/DemoTrd/TransY = 0 cm
d:Ge/DemoTrd/TransZ = 66 cm
d:Ge/DemoTrd/RotX = 0 deg
d:Ge/DemoTrd/RotY = 0 deg
d:Ge/DemoTrd/RotZ = 0 deg
d:Ge/DemoTrd/HLX1 = 30. mm
d:Ge/DemoTrd/HLX2 = 10. mm
d:Ge/DemoTrd/HLY1 = 40. mm
d:Ge/DemoTrd/HLY2 = 15. mm
d:Ge/DemoTrd/HLZ = 60. mm
s:Ge/DemoTrd/Color = "violet"

# RTrap - Right Angular Wedge Trapezoid
s:Ge/DemoRTrap/Type = "G4RTrap"
s:Ge/DemoRTrap/Parent = "World"
s:Ge/DemoRTrap/Material = "Air"
d:Ge/DemoRTrap/TransX = 0 cm
d:Ge/DemoRTrap/TransY = 0 cm
d:Ge/DemoRTrap/TransZ = 84 cm
d:Ge/DemoRTrap/RotX = 0 deg
d:Ge/DemoRTrap/RotY = 0 deg
d:Ge/DemoRTrap/RotZ = 0 deg
d:Ge/DemoRTrap/LZ = 120. mm
d:Ge/DemoRTrap/LY = 80. mm
d:Ge/DemoRTrap/LX = 60. mm
d:Ge/DemoRTrap/LTX = 30. mm
s:Ge/DemoRTrap/Color = "pink"

# GTrap - General Trapezoid
s:Ge/DemoGTrap/Type = "G4GTrap"
s:Ge/DemoGTrap/Parent = "World"
s:Ge/DemoGTrap/Material = "Air"
d:Ge/DemoGTrap/TransX = 0 cm
d:Ge/DemoGTrap/TransY = 0 cm
d:Ge/DemoGTrap/TransZ = 104 cm
d:Ge/DemoGTrap/RotX = 0 deg
d:Ge/DemoGTrap/RotY = 0 deg
d:Ge/DemoGTrap/RotZ = 0 deg
d:Ge/DemoGTrap/HLZ = 60. mm
d:Ge/DemoGTrap/Theta = 20 deg
d:Ge/DemoGTrap/Phi = 5 deg
d:Ge/DemoGTrap/HLY1 = 40. mm
d:Ge/DemoGTrap/HLX1 = 30. mm
d:Ge/DemoGTrap/HLX2 = 40. mm
d:Ge/DemoGTrap/Alp1 = 10 deg
d:Ge/DemoGTrap/HLY2 = 16. mm
d:Ge/DemoGTrap/HLX3 = 10. mm
d:Ge/DemoGTrap/HLX4 = 14. mm
s:Ge/DemoGTrap/Alp2 = 10 deg
s:Ge/DemoGTrap/Color = "indigo"

# Sphere
s:Ge/DemoSphere/Type = "TsSphere"
ss:Ge/DemoSphere/Parent = "World"
ss:Ge/DemoSphere/Material = "Air"
s:Ge/DemoSphere/TransX = 0 cm
s:Ge/DemoSphere/TransY = 0 cm
d:Ge/DemoSphere/TransZ = 124 cm
d:Ge/DemoSphere/RotX = 0 deg
d:Ge/DemoSphere/RotY = 0 deg
d:Ge/DemoSphere/RotZ = 0 deg
d:Ge/DemoSphere/RMin = 100 mm
d:Ge/DemoSphere/RMax = 120 mm
d:Ge/DemoSphere/RPhi = 0 deg
d:Ge/DemoSphere/DPhi = 180 deg
d:Ge/DemoSphere/STheta = 0 deg
d:Ge/DemoSphere/DTheta = 180 deg
d:Ge/DemoSphere/DrawingStyle = "FullWireFrame"
s:Ge/DemoSphere/Color = "grass"

# Orb
s:Ge/DemoOrb/Type = "G4Orb"
ss:Ge/DemoOrb/Parent = "World"
ss:Ge/DemoOrb/Material = "Air"
s:Ge/DemoOrb/TransX = 0 cm
s:Ge/DemoOrb/TransY = 0 cm
d:Ge/DemoOrb/TransZ = 150 cm
d:Ge/DemoOrb/RotX = 0 deg
d:Ge/DemoOrb/RotY = 0 deg
d:Ge/DemoOrb/RotZ = 0 deg
d:Ge/DemoOrb/R = 100 mm
d:Ge/DemoOrb/Color = "orange"
ss:Ge/DemoOrb/DrawingStyle = "FullWireFrame"

# Torus
s:Ge/DemoTorus/Type = "G4Torus"
ss:Ge/DemoTorus/Parent = "World"
ss:Ge/DemoTorus/Material = "Air"
s:Ge/DemoTorus/TransX = 0 cm
s:Ge/DemoTorus/TransY = 0 cm
d:Ge/DemoTorus/TransZ = 176 cm
d:Ge/DemoTorus/RotX = 0 deg
d:Ge/DemoTorus/RotY = 0 deg
d:Ge/DemoTorus/RotZ = 0 deg
d:Ge/DemoTorus/RMin = 40 mm
d:Ge/DemoTorus/RMax = 60 mm
d:Ge/DemoTorus/RTor = 200 mm
d:Ge/DemoTorus/RPhi = 0 deg
d:Ge/DemoTorus/DPhi = 90 deg
d:Ge/DemoTorus/Color = "purple"
ss:Ge/DemoTorus/DrawingStyle = "FullWireFrame"

# HPolycone - Hollow Polycone
s:Ge/DemoHPolycone/Type = "G4HPolycone"
s:Ge/DemoHPolycone/Parent = "World"
# SPolycone - Solid Polycone
s:Ge/DemoSPolycone/Type = "G4SPolycone"
s:Ge/DemoSPolycone/Parent = "World"
s:Ge/DemoSPolycone/Material = "Air"
d:Ge/DemoSPolycone/TransX = 0 cm
d:Ge/DemoSPolycone/TransY = 0 cm
d:Ge/DemoSPolycone/TransZ = 186 cm
d:Ge/DemoSPolycone/RotX = 0 deg
d:Ge/DemoSPolycone/RotY = 0 deg
d:Ge/DemoSPolycone/RotZ = 0 deg
d:Ge/DemoSPolycone/PhiStart = 0.25 rad
d:Ge/DemoSPolycone/PhiTotal = 1.5 rad
dv:Ge/DemoSPolycone/R = 10 0 10 5 10 10 2 2 0 mm
dv:Ge/DemoSPolycone/Z = 10 5 7 9 11 25 27 29 31 35 35 mm
s:Ge/DemoSPolycone/Color = "grey"
s:Ge/DemoSPolycone/DrawingStyle = "FullWireFrame"

# HPolyhedra - Hollow Polyhedra
s:Ge/DemoHPolyhedra/Type = "G4HPolyhedra"
s:Ge/DemoHPolyhedra/Parent = "World"
s:Ge/DemoHPolyhedra/Material = "Air"
d:Ge/DemoHPolyhedra/TransX = 0 cm
d:Ge/DemoHPolyhedra/TransY = 0 cm
d:Ge/DemoHPolyhedra/TransZ = 192 cm
d:Ge/DemoHPolyhedra/RotX = 0 deg
d:Ge/DemoHPolyhedra/RotY = 0 deg
d:Ge/DemoHPolyhedra/RotZ = 0 deg
d:Ge/DemoHPolyhedra/PhiStart = -0.25 rad
d:Ge/DemoHPolyhedra/PhiTotal = 1.25 rad
i:Ge/DemoHPolyhedra/NSides = 3
dv:Ge/DemoHPolyhedra/Z = 7 0 5 8 13 30 32 35 mm
dv:Ge/DemoHPolyhedra/RInner = 7 0 2 2 3 1 1 2 mm
dv:Ge/DemoHPolyhedra/ROuter = 7 0 15 15 4 4 10 10 mm
s:Ge/DemoHPolyhedra/Color = "blue"

# SPolyhedra - Solid Polyhedra
s:Ge/DemoSPolyhedra/Type = "G4SPolyhedra"
s:Ge/DemoSPolyhedra/Parent = "World"
s:Ge/DemoSPolyhedra/Material = "Air"
d:Ge/DemoSPolyhedra/TransX = 0 cm
d:Ge/DemoSPolyhedra/TransY = 0 cm
d:Ge/DemoSPolyhedra/TransZ = 198 cm
d:Ge/DemoSPolyhedra/RotX = 0 deg
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/DemoSPolyhedra/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/PhiStart</td>
<td>-0.25 rad</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/PhiTotal</td>
<td>1.25 rad</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/NSides</td>
<td>100</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/R</td>
<td>5 0 4 12 8 0 mm</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/Z</td>
<td>5 0 0 8 16 16 mm</td>
</tr>
<tr>
<td>Ge/DemoSPolyhedra/Color</td>
<td>&quot;green&quot;</td>
</tr>
</tbody>
</table>

**# EllipticalTube**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/DemoEllipticalTube/Type</td>
<td>&quot;G4EllipticalTube&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/Parent</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/Material</td>
<td>&quot;Air&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/TransY</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/TransZ</td>
<td>204 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/HLX</td>
<td>5 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/HLY</td>
<td>10 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/HLZ</td>
<td>20 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/Color</td>
<td>&quot;yellow&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalTube/DrawingStyle</td>
<td>&quot;FullWireFrame&quot;</td>
</tr>
</tbody>
</table>

**# Ellipsoid**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/DemoEllipsoid/Type</td>
<td>&quot;G4Ellipsoid&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/Parent</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/Material</td>
<td>&quot;Air&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/TransY</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/TransZ</td>
<td>210 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/HLX</td>
<td>10 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/HLY</td>
<td>20 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/HLZ</td>
<td>50 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/ZBottom</td>
<td>-10 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/ZTop</td>
<td>40 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/Color</td>
<td>&quot;white&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipsoid/DrawingStyle</td>
<td>&quot;FullWireFrame&quot;</td>
</tr>
</tbody>
</table>

**# EllipticalCone**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/DemoEllipticalCone/Type</td>
<td>&quot;G4EllipticalCone&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/Parent</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/Material</td>
<td>&quot;Air&quot;</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/TransY</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/TransZ</td>
<td>222 cm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/RMin</td>
<td>10 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/RMax</td>
<td>15 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/HLX</td>
<td>.4 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/HLY</td>
<td>.8 mm</td>
</tr>
<tr>
<td>Ge/DemoEllipticalCone/ZMax</td>
<td>50 mm</td>
</tr>
</tbody>
</table>
**Paraboloid**

- `Ge/DemoParaboloid/Type = "G4Paraboloid"
- `Ge/DemoParaboloid/Parent = "World"
- `Ge/DemoParaboloid/Material = "Air"
- `Ge/DemoParaboloid/TransX = 0 cm
- `Ge/DemoParaboloid/TransY = 0 cm
- `Ge/DemoParaboloid/TransZ = 234 cm
- `Ge/DemoParaboloid/RotX = 0 deg
- `Ge/DemoParaboloid/RotY = 0 deg
- `Ge/DemoParaboloid/RotZ = 0 deg
- `Ge/DemoParaboloid/HLZ = 20 mm
- `Ge/DemoParaboloid/R1 = 20 mm
- `Ge/DemoParaboloid/R2 = 35 mm
- `Ge/DemoParaboloid/Color = "skyblue"
- `Ge/DemoParaboloid/DrawingStyle = "FullWireFrame"

**Hype**

- `Ge/DemoHype/Type = "G4Hype"
- `Ge/DemoHype/Parent = "World"
- `Ge/DemoHype/Material = "Air"
- `Ge/DemoHype/TransX = 0 cm
- `Ge/DemoHype/TransY = 0 cm
- `Ge/DemoHype/TransZ = 248 cm
- `Ge/DemoHype/RotX = 0 deg
- `Ge/DemoHype/RotY = 0 deg
- `Ge/DemoHype/RotZ = 0 deg
- `Ge/DemoHype/IR = 20 mm
- `Ge/DemoHype/OR = 30 mm
- `Ge/DemoHype/IS = .7 rad
- `Ge/DemoHype/OS = .7 rad
- `Ge/DemoHype/HLZ = 50 mm
- `Ge/DemoHype/Color = "red"
- `Ge/DemoHype/DrawingStyle = "FullWireFrame"

**Tet**

- `Ge/DemoTet/Type = "G4Tet"
- `Ge/DemoTet/Parent = "World"
- `Ge/DemoTet/Material = "Air"
- `Ge/DemoTet/TransX = 0 cm
- `Ge/DemoTet/TransY = 0 cm
- `Ge/DemoTet/TransZ = 262 cm
- `Ge/DemoTet/RotX = 0 deg
- `Ge/DemoTet/RotY = 0 deg
- `Ge/DemoTet/RotZ = 0 deg
- `Ge/DemoTet/Anchor = 3 0 0 17.3 mm
- `Ge/DemoTet/P2 = 3 0 16.3 -5.8 mm
- `Ge/DemoTet/P3 = 3 -14.1 -8.2 -5.8 mm
- `Ge/DemoTet/P4 = 3 14.1 -8.2 -5.8 mm
- `Ge/DemoTet/Color = "magenta"

**ExtrudedSolid**

- `Ge/DemoExtrudedSolid/Type = "G4ExtrudedSolid"
- `Ge/DemoExtrudedSolid/Parent = "World"
- `Ge/DemoExtrudedSolid/Material = "Air"
# TwistedBox

```plaintext
s: Ge/DemoG4TwistedBox/Type = "G4TwistedBox"
s: Ge/DemoG4TwistedBox/Parent = "World"
s: Ge/DemoG4TwistedBox/Material = "Air"
d: Ge/DemoG4TwistedBox/TransX = 0 cm
d: Ge/DemoG4TwistedBox/TransY = 0 cm
d: Ge/DemoG4TwistedBox/TransZ = 284 cm
d: Ge/DemoG4TwistedBox/RotX = 0 deg
d: Ge/DemoG4TwistedBox/RotY = 0 deg
d: Ge/DemoG4TwistedBox/RotZ = 0 deg
d: Ge/DemoG4TwistedBox/Twist = 30 deg
d: Ge/DemoG4TwistedBox/HLX = 30. mm
d: Ge/DemoG4TwistedBox/HLY = 40. mm
d: Ge/DemoG4TwistedBox/HLZ = 60. mm
s: Ge/DemoG4TwistedBox/Color = "pink"
s: Ge/DemoG4TwistedBox/DrawingStyle = "FullWireFrame"
```

# RTwistedTrap - Right Angular Wedge Twisted Trapezoid

```plaintext
s: Ge/DemoRTwistedTrap/Type = "G4RTwistedTrap"
s: Ge/DemoRTwistedTrap/Parent = "World"
s: Ge/DemoRTwistedTrap/Material = "Air"
d: Ge/DemoRTwistedTrap/TransX = 0 cm
d: Ge/DemoRTwistedTrap/TransY = 0 cm
d: Ge/DemoRTwistedTrap/TransZ = 302 cm
d: Ge/DemoRTwistedTrap/RotX = 0 deg
d: Ge/DemoRTwistedTrap/RotY = 0 deg
d: Ge/DemoRTwistedTrap/RotZ = 0 deg
d: Ge/DemoRTwistedTrap/Twist = 30 deg
d: Ge/DemoRTwistedTrap/HLX1 = 30. mm
d: Ge/DemoRTwistedTrap/HLX2 = 40. mm
d: Ge/DemoRTwistedTrap/HLY1 = 40. mm
d: Ge/DemoRTwistedTrap/HLY2 = 60. mm
s: Ge/DemoRTwistedTrap/Color = "indigo"
s: Ge/DemoRTwistedTrap/DrawingStyle = "FullWireFrame"
```

# GTwistedTrap - General Twisted Trapezoid

```plaintext
s: Ge/DemoGTwistedTrap/Type = "G4GTwistedTrap"
s: Ge/DemoGTwistedTrap/Parent = "World"
s: Ge/DemoGTwistedTrap/Material = "Air"
d: Ge/DemoGTwistedTrap/TransX = 0 cm
d: Ge/DemoGTwistedTrap/TransY = 0 cm
d: Ge/DemoGTwistedTrap/TransZ = 320 cm
d: Ge/DemoGTwistedTrap/RotX = 0 deg
```
d:Ge/DemoGTwistedTrap/RotY = 0 deg

d:Ge/DemoGTwistedTrap/RotZ = 0 deg

d:Ge/DemoGTwistedTrap/Twist = 30 deg

d:Ge/DemoGTwistedTrap/HLZ = 60. mm

d:Ge/DemoGTwistedTrap/Theta = 20 deg

d:Ge/DemoGTwistedTrap/Phi = 5 deg

d:Ge/DemoGTwistedTrap/HLY1 = 40. mm

d:Ge/DemoGTwistedTrap/HLX1 = 30. mm

d:Ge/DemoGTwistedTrap/HLX2 = 40. mm

d:Ge/DemoGTwistedTrap/HLY2 = 16. mm

d:Ge/DemoGTwistedTrap/HLX3 = 10. mm

d:Ge/DemoGTwistedTrap/HLX4 = 14. mm

d:Ge/DemoGTwistedTrap/Alpha = 10 deg

d:Ge/DemoGTwistedTrap/Color = "grass"

d:Ge/DemoGTwistedTrap/DrawingStyle = "FullWireFrame"

# TwistedTrd

d:Ge/DemoTwistedTrd/Type = "G4TwistedTrd"

d:Ge/DemoTwistedTrd/Parent = "World"

d:Ge/DemoTwistedTrd/Material = "Air"

d:Ge/DemoTwistedTrd/TransX = 0 cm

d:Ge/DemoTwistedTrd/TransY = 0 cm

d:Ge/DemoTwistedTrd/TransZ = 336 cm

d:Ge/DemoTwistedTrd/RotX = 0 deg

d:Ge/DemoTwistedTrd/RotY = 0 deg

d:Ge/DemoTwistedTrd/RotZ = 0 deg

d:Ge/DemoTwistedTrd/HLX1 = 30. mm

d:Ge/DemoTwistedTrd/HLX2 = 10. mm

d:Ge/DemoTwistedTrd/HLY1 = 40. mm

d:Ge/DemoTwistedTrd/HLY2 = 15. mm

d:Ge/DemoTwistedTrd/HLZ = 60. mm

d:Ge/DemoTwistedTrd/Twist = 30 deg

d:Ge/DemoTwistedTrd/Color = "orange"

d:Ge/DemoTwistedTrd/DrawingStyle = "FullWireFrame"

# GenericTrap

d:Ge/DemoGenericTrap/Type = "G4GenericTrap"

d:Ge/DemoGenericTrap/Parent = "World"

d:Ge/DemoGenericTrap/Material = "Air"

d:Ge/DemoGenericTrap/TransX = 0 cm

d:Ge/DemoGenericTrap/TransY = 0 cm

d:Ge/DemoGenericTrap/TransZ = 350 cm

d:Ge/DemoGenericTrap/RotX = 0 deg

d:Ge/DemoGenericTrap/RotY = 0 deg

d:Ge/DemoGenericTrap/RotZ = 0 deg

d:Ge/DemoGenericTrap/HLZ = 25 mm

d:Ge/DemoGenericTrap/Vertices = 16 -30 -30 -30 30 30 30 30 -30 -5 -20 -20 20 20 20 -20 mm

d:Ge/DemoGenericTrap/Color = "purple"

d:Ge/DemoGenericTrap/DrawingStyle = "FullWireFrame"

# TwistedTubs

d:Ge/DemoTwistedTubs/Type = "G4TwistedTubs"

d:Ge/DemoTwistedTubs/Parent = "World"

d:Ge/DemoTwistedTubs/Material = "Air"

d:Ge/DemoTwistedTubs/TransX = 0 cm

d:Ge/DemoTwistedTubs/TransY = 0 cm

d:Ge/DemoTwistedTubs/TransZ = 358 cm
# Demonstrates all of the standard Geant4 solids.
# Shows only the required parameters, omitting the optional ones
# (the ones such as TransX or RMin that have default values).
# Details are given in the TOPAS User Guide
# and in chapter 4 of the Geant4 Application Developer's Guide

# Box
s:Ge/DemoBox/Type = "TsBox"
s:Ge/DemoBox/Parent = "World"
s:Ge/DemoBox/Material = "Air"
d:Ge/DemoBox/HLX = 30. mm
d:Ge/DemoBox/HLY = 40. mm
d:Ge/DemoBox/HLZ = 60. mm
s:Ge/DemoBox/Color = "white"

# Tubs
s:Ge/DemoCylinder/Type = "TsCylinder"
s:Ge/DemoCylinder/Parent = "World"
s:Ge/DemoCylinder/Material = "Air"
d:Ge/DemoCylinder/TransZ = 12 cm
d:Ge/DemoCylinder/RMax = 15 mm
d:Ge/DemoCylinder/HL = 20 mm
s:Ge/DemoCylinder/Color = "lightblue"
s:Ge/DemoCylinder/DrawingStyle = "FullWireFrame"

# CutTubs
s:Ge/DemoCutTubs/Type = "G4CutTubs"
s:Ge/DemoCutTubs/Parent = "World"
s:Ge/DemoCutTubs/Material = "Air"
s:Ge/DemoCutTubs/TransZ = 24 cm
s:Ge/DemoCutTubs/RMax = 20 mm
s:Ge/DemoCutTubs/HL = 30 mm
uv:Ge/DemoCutTubs/LowNorm = 3 0. -0.7 -0.71
uv:Ge/DemoCutTubs/HighNorm = 3 0.7 0. 0.71
s:Ge/DemoCutTubs/Color = "skyblue"
# Cons
s:Ge/DemoCons/Type = "G4Cons"
s:Ge/DemoCons/Parent = "World"
s:Ge/DemoCons/Material = "Air"
s:Ge/DemoCons/TransZ = 36 cm
d:Ge/DemoCons/RMax1 = 10 mm
d:Ge/DemoCons/RMax2 = 25 mm
d:Ge/DemoCons/HL = 40 mm
s:Ge/DemoCons/Color = "red"
s:Ge/DemoCons/DrawingStyle = "FullWireFrame"

# Para
s:Ge/DemoPara/Type = "G4Para"
s:Ge/DemoPara/Parent = "World"
s:Ge/DemoPara/Material = "Air"

d:Ge/DemoPara/TransX = 0 cm
d:Ge/DemoPara/TransY = 0 cm
d:Ge/DemoPara/TransZ = 48 cm
d:Ge/DemoPara/RotX = 0 deg
d:Ge/DemoPara/RotY = 0 deg
d:Ge/DemoPara/RotZ = 0 deg
d:Ge/DemoPara/HLX = 30. mm
d:Ge/DemoPara/HLY = 40. mm
d:Ge/DemoPara/HLZ = 60. mm
d:Ge/DemoPara/Alpha = 0.3 rad
d:Ge/DemoPara/Theta = 0 rad
d:Ge/DemoPara/Phi = 0 rad
s:Ge/DemoPara/Color = "magenta"

# Trd
s:Ge/DemoTrd/Type = "G4Trd"
s:Ge/DemoTrd/Parent = "World"
s:Ge/DemoTrd/Material = "Air"


d:Ge/DemoTrd/TransX = 0 cm
d:Ge/DemoTrd/TransY = 0 cm
d:Ge/DemoTrd/TransZ = 66 cm
d:Ge/DemoTrd/RotX = 0 deg
d:Ge/DemoTrd/RotY = 0 deg
d:Ge/DemoTrd/RotZ = 0 deg
d:Ge/DemoTrd/HLX1 = 30. mm
d:Ge/DemoTrd/HLX2 = 10. mm
d:Ge/DemoTrd/HLY1 = 40. mm
d:Ge/DemoTrd/HLY2 = 15. mm
d:Ge/DemoTrd/HLZ = 60. mm
s:Ge/DemoTrd/Color = "violet"

# RTrap - Right Angular Wedge Trapezoid
s:Ge/DemoRTrap/Type = "G4RTrap"
s:Ge/DemoRTrap/Parent = "World"
s:Ge/DemoRTrap/Material = "Air"

d:Ge/DemoRTrap/TransX = 0 cm
d:Ge/DemoRTrap/TransY = 0 cm
d:Ge/DemoRTrap/TransZ = 84 cm
d:Ge/DemoRTrap/RotX = 0 deg
d:Ge/DemoRTrap/RotY = 0 deg
d:Ge/DemoRTrap/RotZ = 0 deg
TOPAS Documentation, Release 3.0

Ellipsoid

```plaintext
>G4Ellipsoid

Parent = "World"
Material = "Air"
TransZ = 210 cm
HLX = 10 mm
HLY = 20 mm
HLZ = 50 mm
Color = "white"
DrawingStyle = "FullWireFrame"
```

EllipticalCone

```plaintext
>G4EllipticalCone

Parent = "World"
Material = "Air"
TransZ = 222 cm
HLX = .4 mm
HLY = .8 mm
ZMax = 50 mm
Color = "lightblue"
DrawingStyle = "FullWireFrame"
```

Paraboloid

```plaintext
>G4Paraboloid

Parent = "World"
Material = "Air"
TransZ = 234 cm
HLZ = 20 mm
R1 = 20 mm
R2 = 35 mm
Color = "skyblue"
DrawingStyle = "FullWireFrame"
```

Hype

```plaintext
>G4Hype

Parent = "World"
Material = "Air"
TransZ = 248 cm
OR = 30 mm
OS = .7 rad
HLZ = 50 mm
Color = "red"
DrawingStyle = "FullWireFrame"
```

Tet

```plaintext
>G4Tet

Parent = "World"
Material = "Air"
TransZ = 262 cm
Anchor = 3 0 0 17.3 mm
P2 = 3 0 16.3 -5.8 mm
P3 = 3 -14.1 -8.2 -5.8 mm
P4 = 3 14.1 -8.2 -5.8 mm
Color = "magenta"
```

ExtrudedSolid

```plaintext
>G4ExtrudedSolid

Parent = "World"
```
# Demonstrates use of EnergySpectrum.
# Resulting particles can be seen in ASCIIOutput.phsp

So/Example/BeamEnergySpectrumType = "Continuous"
So/Example/BeamEnergySpectrumValues = 3 50. 100. 150. MeV
So/Example/BeamEnergySpectrumWeights = 3 .20 .60 .20
TwissPrimary.txt

# Demonstrates use of Twiss parameterization

Ts/TrackingVerbosity = 0

#---- Sequence
Ge/CheckForUnusedComponents = "False"
Ts/ShowCPUTime = "true"
Ts/ShowHistoryCountAtInterval = 1000000
Ts/Seed = 10

#---- Constants
d:Ge/MyBeamSizeInX = 5.0 cm
d:Ge/MyBeamSizeInY = 5.0 cm

#--- Physics
sv:Ph/Default/Modules =1 "g4em-standard_opt0"

#------ Primary: Twiss parameterization
s:So/Default/Type = "twiss"
s:So/Default/Component = "PlaneAtBeamPosition"
s:So/Default/BeamParticle = "proton"
s:So/Default/BeamEnergy = 150.00 MeV
u:So/Default/BeamEnergySpread = 1.0
i:So/Default/NumberOfHistoriesInRun = 10000

i:Ts/ShowHistoryCountAtInterval = 10
- So/Default/AlphaOfX = 1.135
- So/Default/BetaOfX = 7.40 cm
- So/Default/EmittanceOfX = 1.2299 cm
- So/Default/SigmaOfX = 0.2 cm
- So/Default/SigmaOfXPrime = 0.01
- So/Default/AlphaOfY = 1.135
- So/Default/BetaOfY = -7.407 cm
- So/Default/EmittanceOfY = 1.2299 cm
- So/Default/SigmaOfY = 0.2 cm
- So/Default/SigmaOfYPrime = 0.03

--- Beam Box
Ge/PlaneAtBeamPosition/Type = "G4Box"
Ge/PlaneAtBeamPosition/Material = "Vacuum"
Ge/PlaneAtBeamPosition/Parent = "World"
Ge/PlaneAtBeamPosition/HLX = 0.5 cm + Ge/MyBeamSizeInX
Ge/PlaneAtBeamPosition/HLY = 0.5 cm + Ge/MyBeamSizeInY
Ge/PlaneAtBeamPosition/HLZ = 1.0 mm
Ge/PlaneAtBeamPosition/TransZ = 1.0 m
Ge/PlaneAtBeamPosition/TransY = 0. m
Ge/PlaneAtBeamPosition/TransX = 0. m
Ge/PlaneAtBeamPosition/RotX = 0. deg
Ge/PlaneAtBeamPosition/RotY = 180. deg
Ge/PlaneAtBeamPosition/RotZ = 0. deg

--- Phase space scoring at Beam geometry
Sc/PsPrimary_0_Beam/Quantity = "PhaseSpace"
Sc/PsPrimary_0_Beam/OutputToConsole = "True"
Sc/PsPrimary_0_Beam/Surface = "PlaneAtBeamPosition/ZPlusSurface"
Sc/PsPrimary_0_Beam/OutputType = "ASCII"
Sc/PsPrimary_0_Beam/PhaseSpaceBufferSize = 1000
Sc/PsPrimary_0_Beam/IncludeRunID = "True"
Sc/PsPrimary_0_Beam/IncludeEventID = "True"
Sc/PsPrimary_0_Beam/IncludeTrackID = "True"
Sc/PsPrimary_0_Beam/IncludeTime = "True"
Sc/PsPrimary_0_Beam/IncludeSeed = "True"
Sc/PsPrimary_0_Beam/IfOutputFileAlreadyExists = "Overwrite"

--- Target box the ZPlus surface at Iso center
Ge/PlaneAtIsoCenter/Type = "G4Box"
Ge/PlaneAtIsoCenter/Material = "Vacuum"
Ge/PlaneAtIsoCenter/Parent = "World"
Ge/PlaneAtIsoCenter/HLX = 2 * Ge/PlaneAtBeamPosition/HLX cm
Ge/PlaneAtIsoCenter/HLY = 2 * Ge/PlaneAtBeamPosition/HLY cm
Ge/PlaneAtIsoCenter/HLZ = 0.5 mm
Ge/PlaneAtIsoCenter/TransZ = -1.0 * Ge/PlaneAtIsoCenter/HLZ cm
Ge/PlaneAtIsoCenter/TransY = 0. m
Ge/PlaneAtIsoCenter/TransX = 0. m
Ge/PlaneAtIsoCenter/RotX = 0. deg
Ge/PlaneAtIsoCenter/RotY = 0. deg
Ge/PlaneAtIsoCenter/RotZ = 0. deg

--- Phase space scoring at Iso center
Sc/PsPrimary_0_Iso/Quantity = "PhaseSpace"
Sc/PsPrimary_0_Iso/OutputToConsole = "True"
Sc/PsPrimary_0_Iso/Surface = "PlaneAtIsoCenter/ZPlusSurface"
### TwoBeams.txt

# Demonstrates ability to have more than one particle source

a:Ge/World/Material = "Vacuum"

# Second beam component

d:Ge/XRayTube/Parent = "World"
d:Ge/XRayTube/Type = "Group"
d:Ge/XRayTube/TransX = 200. cm
d:Ge/XRayTube/TransY = 0. m
d:Ge/XRayTube/TransZ = 0. m
d:Ge/XRayTube/RotX = 0. deg
d:Ge/XRayTube/RotY = 90. deg
d:Ge/XRayTube/RotZ = 0. deg

# Second beam source

d:So/Imaging/Type = "Beam"
d:So/Imaging/Component = "XRayTube"
d:So/Imaging/BeamParticle = "gamma"
d:So/Imaging/BeamEnergy = 100. keV
d:So/Imaging/BeamEnergySpread = 0.
d:So/Imaging/BeamPositionCutoffShape = "Ellipse"
d:So/Imaging/BeamPositionDistribution = "Flat"
d:So/Imaging/BeamPositionCutoffX = 7. cm
d:So/Imaging/BeamPositionCutoffY = 21. cm
d:So/Imaging/BeamAngularDistribution = "None"
d:So/Imaging/NumberOfHistoriesInRun = 5

a:So/Therapy/Type = "Beam"
a:So/Therapy/Component = "BeamPosition"
a:So/Therapy/BeamParticle = "proton"
d:So/Therapy/BeamEnergy = 169.23 MeV
d:So/Therapy/BeamEnergySpread = 0.757504
d:So/Therapy/BeamPositionDistribution = "Gaussian"
da:So/Therapy/BeamPositionCutoffShape = "Ellipse"
da:So/Therapy/BeamPositionCutoffX = 10. cm
da:So/Therapy/BeamPositionCutoffY = 10. cm
d:So/Therapy/BeamPositionSpreadX = 0.65 cm
da:So/Therapy/BeamPositionSpreadY = 0.65 cm
da:So/Therapy/BeamAngularDistribution = "Gaussian"
da:So/Therapy/BeamAngularCutoffX = 90. deg
da:So/Therapy/BeamAngularCutoffY = 90. deg
da:So/Therapy/BeamAngularSpreadX = 0.0032 rad
da:So/Therapy/BeamAngularSpreadY = 0.0032 rad

d:So/Therapy/NumberOfHistoriesInRun = 10
s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
b:Gr/ViewA/IncludeAxes = "True"
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 2.

b:Ts/PauseBeforeQuit = "True"
ColorByOriginComponent.txt

# Demonstrates changing color scheme of trajectories.
# Trajectories that originate in the component named Outer
# or any of its subcomponents will be colored red.
# All others will be colored grey.

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 4. m
d:Ge/Box/HLY = 4. m
d:Ge/Box/HLZ = .4 m
d:Ge/Box/TransX = 0. m
d:Ge/Box/TransY = 0. m
d:Ge/Box/TransZ = -2. m
d:Ge/Box/RotX = 0. deg
d:Ge/Box/RotY = 0. deg
d:Ge/Box/RotZ = 0. deg

d:Ge/Outer/Type = "TsBox"
s:Ge/Outer/Parent = "World"
s:Ge/Outer/Material = "G4_WATER"
d:Ge/Outer/HLX = 2.01 m
d:Ge/Outer/HLY = 2.01 m
d:Ge/Outer/HLZ = .11 m
d:Ge/Outer/TransX = 0. m
d:Ge/Outer/TransY = 0. m
d:Ge/Outer/TransZ = 2. m
d:Ge/Outer/RotX = 0. deg
d:Ge/Outer/RotY = 0. deg
d:Ge/Outer/RotZ = 0. deg
```plaintext
s:Ge/Film1/Type = "TsBox"
s:Ge/Film1/Parent = "Outer"
s:Ge/Film1/Material = "G4_WATER"
d:Ge/Film1/HLX = 2. m
d:Ge/Film1/HLY = 2. m
d:Ge/Film1/HLZ = .1 m
d:Ge/Film1/TransX = 0. m
d:Ge/Film1/TransY = 0. m
d:Ge/Film1/TransZ = 0. m
d:Ge/Film1/RotX = 0. deg
d:Ge/Film1/RotY = 0. deg
d:Ge/Film1/RotZ = 0. deg
i:Ge/Film1/XBins = 2
i:Ge/Film1/YPins = 2
i:Ge/Film1/ZBins = 2

s:Ge/Film2/Type = "TsBox"
s:Ge/Film2/Parent = "World"
s:Ge/Film2/Material = "G4_WATER"
d:Ge/Film2/HLX = 3. m
d:Ge/Film2/HLY = 3. m
d:Ge/Film2/HLZ = .1 m
d:Ge/Film2/TransX = 0. m
d:Ge/Film2/TransY = 0. m
d:Ge/Film2/TransZ = 0. m
d:Ge/Film2/RotX = 0. deg
d:Ge/Film2/RotY = 0. deg
d:Ge/Film2/RotZ = 0. deg

s:Gr/MyOGL/Type = "OpenGL"
i:Gr/MyOGL/WindowSizeX = 600
i:Gr/MyOGL/WindowSizeY = 600
i:Gr/MyOGL/WindowPosX = 0
i:Gr/MyOGL/WindowPosY = 0
s:Gr/MyOGL/ColorBy = "OriginComponentOrSubComponentOf"
v:Gr/MyOGL/ColorByOriginComponentNames = 1 "Outer"
v:Gr/MyOGL/ColorByOriginComponentColors = 1 "Red"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 400 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

D:Ts/PauseBeforeQuit = "True"
```
SolidSphere.txt

# Demonstrate effect of drawing style and hidden line removal

s:Ge/Sphere/Type = "TsSphere"
s:Ge/Sphere/Parent = "World"
s:Ge/Sphere/Material = "G4_WATER"
d:Ge/Sphere/RMin = 0. m
d:Ge/Sphere/RMax = 2. m
d:Ge/Sphere/SPhi = 0. deg
d:Ge/Sphere/DPhi = 360. deg
d:Ge/Sphere/STheta = 0. deg
d:Ge/Sphere/DTheta = 180. deg
d:Ge/Sphere/TransX = 0. m
d:Ge/Sphere/TransY = 0. m
d:Ge/Sphere/TransZ = 0. m
d:Ge/Sphere/RotX = 0. deg
d:Ge/Sphere/RotY = 0. deg
d:Ge/Sphere/RotZ = 0. deg
s:Ge/Sphere/DrawingStyle = "Solid"

d:Gr/MyOGL/Type = "OpenGL"
i:Gr/MyOGL/WindowSizeX = 600
i:Gr/MyOGL/WindowSizeY = 600
i:Gr/MyOGL/WindowPosX = 0
i:Gr/MyOGL/WindowPosY = 0
d:Gr/MyOGL/Theta = 30. deg
d:Gr/MyOGL/Phi = 30. deg
u:Gr/MyOGL/Zoom = 3.
s:Gr/MyOGL/ColorBy = "particletype"
b:Gr/MyOGL/HiddenLineRemovalForTrajectories = "True"

b:So/Example/Type = "Beam"
\d:So/Example/Component = "BeamPosition"
\d:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
\d:So/Example/BeamPositionDistribution = "Gaussian"
\d:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
\d:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 100

b:Ts/PauseBeforeQuit = "True"
# Demonstrates a variety of Graphics options

```plaintext
s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
s:Ge/Box/DrawingStyle = "Solid"

s:Gr/MyOGL/Type = "OpenGL"
b:Gr/MyOGL/CopyOpenGLToEPS = "True"
i:Gr/MyOGL/WindowSizeX = 600
i:Gr/MyOGL/WindowSizeY = 600
i:Gr/MyOGL/WindowPosX = 0
i:Gr/MyOGL/WindowPosY = 0
u:Gr/MyOGL/Zoom = 3.
d:Gr/MyOGL/Theta = 30. deg
d:Gr/MyOGL/Phi = 30. deg
b:Gr/MyOGL/IncludeGeometry = "t"
b:Gr/MyOGL/IncludeTrajectories = "t"
b:Gr/MyOGL/HiddenLineRemovalForGeometry = "f"
b:Gr/MyOGL/HiddenLineRemovalForTrajectories = "f"

# ColorBy options are "Charge", "ParticleType", "OriginComponent", "Energy",
# "Momentum", "Generation" and "CreatorProcess"
s:Gr/MyOGL/ColorBy = "particletype"
sv:Gr/MyOGL/ColorByChargeColors = 3 "red" "red" "red"
sv:Gr/MyOGL/ColorByParticleTypeNames = 2 "proton" "e-"
sv:Gr/MyOGL/ColorByParticleTypeColors = 2 "red" "green"
sv:Gr/MyOGL/ColorByOriginComponentNames = 2 "World" "Box"
sv:Gr/MyOGL/ColorByOriginComponentColors = 2 "yellow" "red"
dv:Gr/MyOGL/ColorByEnergyRanges = 4 .3 2. 100. 200. MeV
sv:Gr/MyOGL/ColorByEnergyColors = 5 "red" "yellow" "green" "blue" "white"
dv:Gr/MyOGL/ColorByMomentumRanges = 4 .3 2. 200. 646. MeV
sv:Gr/MyOGL/ColorByMomentumColors = 5 "red" "yellow" "green" "blue" "white"
sv:Gr/MyOGL/ColorByCreatorProcessNames = 5 "eBrem" "annihil" "Decay" "eIoni" "hIoni"
sv:Gr/MyOGL/ColorByCreatorProcessColors = 5 "red" "green" "blue" "yellow" "magenta"
```

a:Gr/MyHepRep/Type = "HepRep"
a:Gr/MyHepRep/FileName = "Above200MeV"

#s:Gr/MyVRML/Type = "VRML"

#s:Gr/MyRayTracer/Type = "RayTracer"

#s:Gr/MyRayTracerX/Type = "RayTracerX"
i:Gr/MyRayTracerX/WindowSizeX = 200
i:Gr/MyRayTracerX/WindowSizeY = 800
i:Gr/MyRayTracerX/WindowPosX = 100
i:Gr/MyRayTracerX/WindowPosY = 300

#s:Gr/MyDAWN/Type = "DAWN"

s:Gr/RefreshEvery = "Run"
#sv:Gr/OnlyIncludeParticlesNamed = 1 "proton"
TwoProjections.txt

# Demonstrate two different kinds of graphical projections.

s:Gr/MyOGLa/Type = "OpenGL"
i:Gr/MyOGLa/WindowSizeX = 400
i:Gr/MyOGLa/WindowSizeY = 400
i:Gr/MyOGLa/WindowPosX = 0
i:Gr/MyOGLa/WindowPosY = 0
s:Gr/MyOGLa/Projection = "Orthogonal"
d:Gr/MyOGLa/Theta = 45. deg
d:Gr/MyOGLa/Phi = 45. deg

s:Gr/MyOGLb/Type = "OpenGL"
i:Gr/MyOGLb/WindowSizeX = 400
i:Gr/MyOGLb/WindowSizeY = 400
i:Gr/MyOGLb/WindowPosX = 0
i:Gr/MyOGLb/WindowPosY = 0
s:Gr/MyOGLb/Projection = "Perspective"
d:Gr/MyOGLb/PerspectiveAngle = 10. deg
d:Gr/MyOGLb/Theta = 45. deg
d:Gr/MyOGLb/Phi = 45. deg

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
d:Ge/Box/TransX = 0. m
d:Ge/Box/TransY = 0. m  
d:Ge/Box/TransZ = 0. m  
d:Ge/Box/RotX = 0. deg  
d:Ge/Box/RotY = 0. deg  
d:Ge/Box/RotZ = 0. deg  

s:So/Example/Type = "Beam"  
s:So/Example/Component = "BeamPosition"  
s:So/Example/BeamParticle = "proton"  
d:So/Example/BeamEnergy = 200. MeV  
u:So/Example/BeamEnergySpread = 0.757504  
s:So/Example/BeamPositionDistribution = "Gaussian"  
s:So/Example/BeamPositionCutoffShape = "Ellipse"  
d:So/Example/BeamPositionCutoffX = 10. cm  
d:So/Example/BeamPositionCutoffY = 10. cm  
d:So/Example/BeamPositionSpreadX = 0.65 cm  
d:So/Example/BeamPositionSpreadY = 0.65 cm  
s:So/Example/BeamAngularDistribution = "Gaussian"  
d:So/Example/BeamAngularCutoffX = 90. deg  
d:So/Example/BeamAngularCutoffY = 90. deg  
d:So/Example/BeamAngularSpreadX = 0.0032 rad  
d:So/Example/BeamAngularSpreadY = 0.0032 rad  
i:So/Example/NumberOfHistoriesInRun = 10  

b:Ts/PauseBeforeQuit = "True"
# Score charge in a sphere

```plaintext
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"
s:Ge/Sphere/Type = "TsSphere"
s:Ge/Sphere/Parent = "World"
s:Ge/Sphere/Material = "Air"
d:Ge/Sphere/RMax = 55.0 cm
s:Ge/Sphere/DrawingStyle = "FullWireFrame"
s:Sc/Charge/Quantity = "Charge"
s:Sc/Charge/Component = "Sphere"
b:Sc/Charge/OutputToConsole = "TRUE"
s:Sc/Charge/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/Charge/OnlyIncludeParticlesNamed = 1 "proton"
#s:Sc/Charge/OnlyIncludeParticlesOfGeneration = "secondary"
s:Gr/ViewA/Type = "OpenGL"
```

```plaintext
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
d:So/Example/BeamEnergySpread = 0.757504
d:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
```
Complex.txt

A complex example showing lots of different scorers and options.

d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"

s:Ge/Film/Type = "TsBox"
s:Ge/Film/Parent = "World"
s:Ge/Film/Material = "G4_WATER"
d:Ge/Film/HLX = 50.0 cm
d:Ge/Film/HLY = 50.0 cm
d:Ge/Film/HLZ = 1.0 cm
d:Ge/Film/TransX = 0. cm
d:Ge/Film/TransY = 0. cm
d:Ge/Film/TransZ = 0. cm
d:Ge/Film/RotX = 0. deg
d:Ge/Film/RotY = 0. deg
d:Ge/Film/RotZ = 0. deg
s:Ge/Film/Color = "skyblue"

s:Ge/Phantom1/Type = "TsBox"
s:Ge/Phantom1/Parent = "World"
s:Ge/Phantom1/Material = "G4_WATER"
d:Ge/Phantom1/HLX = 30.0 cm
d:Ge/Phantom1/HLY = 30.0 cm
d:Ge/Phantom1/HLZ = 1.1 cm
d:Ge/Phantom1/TransZ = -30. cm

s:Ge/Phantom2/Type = "TsBox"
s:Ge/Phantom2/Parent = "World"
s:Ge/Phantom2/Material = "G4_WATER"
d:Ge/Phantom2/HLX = 10.0 cm
d:Ge/Phantom2/HLY = 10.0 cm
d:Ge/Phantom2/HLZ = 10.0 cm
d:Ge/Phantom2/TransZ = -50. cm
s:Ge/Phantom2/Color = "red"
i:Ge/Phantom2/XBins = 2
i:Ge/Phantom2/YBins = 2
i:Ge/Phantom2/ZBins = 1
$a$: Sc/PhaseSpaceAtFilm/Quantity = "PhaseSpace"
$a$: Sc/PhaseSpaceAtFilm/OutputToConsole = "True"
$a$: Sc/PhaseSpaceAtFilm/Surface = "Film/ZMinusSurface"
$a$: Sc/PhaseSpaceAtFilm/OutputType = "ASCII" # ASCII or Binary
$i$: Sc/PhaseSpaceAtFilm/OutputBufferSize = 1000
$\#a$: Sc/PhaseSpaceAtFilm/OnlyIncludeParticlesGoing = "In"
$\#b$: Sc/PhaseSpaceAtFilm/IncludeRunID = "True"
$\#b$: Sc/PhaseSpaceAtFilm/IncludeEventID = "True"
$\#b$: Sc/PhaseSpaceAtFilm/IncludeTrackID = "True"
$\#b$: Sc/PhaseSpaceAtFilm/IncludeTime = "True"
$\#sv$: Sc/PhaseSpaceAtFilm/OnlyIncludeParticlesNamed = 1 "Proton"
$a$: Sc/PhaseSpaceAtFilm/IfOutputFileAlreadyExists = "Increment" # Exit, Overwrite or Increment
$\#a$: Sc/PhaseSpaceAtFilmb/Quantity = "PhaseSpace"
$\#b$: Sc/PhaseSpaceAtFilmb/OutputToConsole = "True"
$\#a$: Sc/PhaseSpaceAtFilmb/Surface = "Film/ZMinusSurface"
$\#a$: Sc/PhaseSpaceAtFilmb/OutputType = "binary" # ASCII or Binary
$i$: Sc/PhaseSpaceAtFilmb/OutputBufferSize = 1000
$\#a$: Sc/PhaseSpaceAtFilmb/OnlyIncludeParticlesGoing = "In"
$\#sv$: Sc/PhaseSpaceAtFilmb/OnlyIncludeParticlesNamed = 1 "Proton"
$a$: Sc/DoseAtPhantom1/Quantity = "DoseToMedium"
$a$: Sc/DoseAtPhantom1/Component = "Phantom1"
$a$: Sc/DoseAtPhantom1/OutputFile = "DoseAtPhantom1"
$a$: Sc/DoseAtPhantom1/OutputType = "csv"
$\#b$: Sc/DoseAtPhantom1/OutputToConsole = "TRUE"
$\#b$: Sc/DoseAtPhantom1/Visualize = "TRUE"
$a$: Sc/DoseAtPhantom1/IfOutputFileAlreadyExists = "Increment" # Exit, Overwrite or Increment
$\#sv$: Sc/DoseAtPhantom1/Report = 8 "Sum" "Mean" "Count_in_bin" "Second_Moment" "Variance"
$\#sv$: Sc/DoseAtPhantom1b/Report = 8 "Sum" "Mean" "Count_in_bin" "Second_Moment" "Variance"
$i$: Sc/DoseAtPhantom1/ZBins = 10
$a$: Sc/DoseAtPhantom1b/Quantity = "DoseToMedium"
$a$: Sc/DoseAtPhantom1b/Component = "Phantom1"
$a$: Sc/DoseAtPhantom1b/OutputFile = "DoseAtPhantom1b"
$a$: Sc/DoseAtPhantom1b/OutputType = "binary"
$\#b$: Sc/DoseAtPhantom1b/OutputToConsole = "TRUE"
$\#b$: Sc/DoseAtPhantom1b/Visualize = "TRUE"
$a$: Sc/DoseAtPhantom1b/IfOutputFileAlreadyExists = "Increment" # Exit, Overwrite or Increment
$\#sv$: Sc/DoseAtPhantom1b/Report = 8 "Sum" "Mean" "Count_in_bin" "Second_Moment" "Variance"
$\#sv$: Sc/EnergyDepAtPhantom1/Quantity = "EnergyDeposit"
$a$: Sc/EnergyDepAtPhantom1/Component = "Phantom1"
$a$: Sc/EnergyDepAtPhantom1/OutputType = "csv"
$\#b$: Sc/EnergyDepAtPhantom1/OutputToConsole = "TRUE"
$\#b$: Sc/EnergyDepAtPhantom1/Visualize = "TRUE"
$a$: Sc/EnergyDepAtPhantom1/IfOutputFileAlreadyExists = "Increment" # Exit, Overwrite or Increment
$\#sv$: Sc/EnergyDepAtPhantom1b/Report = 8 "Sum" "Mean" "Count_in_bin" "Second_Moment" "Variance"
$\#sv$: Sc/FluenceAtPhantom1/Quantity = "Fluence"
$a$: Sc/FluenceAtPhantom1/Component = "Phantom1"
$a$: Sc/FluenceAtPhantom1/OutputFile = "FluenceAtPhantom1"
$a$: Sc/FluenceAtPhantom1/OutputType = "csv"
$b$:Sc/FluenceAtPhantom1/OutputToConsole = "TRUE"
$b$:Sc/FluenceAtPhantom1/Visualize = "TRUE"

$s$:Sc/EnergyFluenceAtPhantom1/Quantity = "EnergyFluence"
$s$:Sc/EnergyFluenceAtPhantom1/Component = "Phantom1"
$s$:Sc/EnergyFluenceAtPhantom1/OutputFile = "EnergyFluenceAtPhantom1"
$s$:Sc/EnergyFluenceAtPhantom1/OutputType = "csv"
$b$:Sc/EnergyFluenceAtPhantom1/OutputToConsole = "TRUE"
$b$:Sc/EnergyFluenceAtPhantom1/Visualize = "TRUE"
$s$:Sc/EnergyFluenceAtPhantom1/OnlyIncludeParticlesNamed = 1 "proton"

$s$:Sc/eDoseAtPhantom1/Quantity = "DoseToMedium"
$s$:Sc/eDoseAtPhantom1/Component = "Phantom1"
$s$:Sc/eDoseAtPhantom1/OnlyIncludeParticlesNamed = 1 "e-
# $s$:Sc/eDoseAtPhantom1/OnlyIncludeParticlesCharged = 1 "negative" # zero or more of positive, negative or neutral
# $d$:Sc/eDoseAtPhantom1/OnlyIncludeIfIncidentParticleKEAbove = 200. keV # minimum energy
# $d$:Sc/eDoseAtPhantom1/OnlyIncludeIfIncidentParticleKEBelow = 250. keV # minimum energy
# $sv$:Sc/eDoseAtPhantom1/OnlyIncludeParticlesFromComponent = 1 "Phantom1"
# $sv$:Sc/eDoseAtPhantom1/OnlyIncludeParticlesFromProcess = 1 "hIoni"
$b$:Sc/eDoseAtPhantom1/InvertFilter = "TRUE"
$s$:Sc/eDoseAtPhantom1/OutputFile = "eDoseAtPhantom1"
$s$:Sc/eDoseAtPhantom1/OutputType = "csv"
$b$:Sc/eDoseAtPhantom1/OutputToConsole = "FALSE"
$b$:Sc/eDoseAtPhantom1/Visualize = "TRUE"

$s$:Sc/EDepAtPhantom1/Quantity = "DoseToMedium"
$s$:Sc/EDepAtPhantom1/Component = "Phantom2"
$b$:Sc/EDepAtPhantom1/OutputToConsole = "TRUE"
$s$:Sc/EDepAtPhantom1/OnlyIncludeParticlesNamed = 1 "proton"

$s$:Sc/DoseAtPhantom2/Quantity = "DoseToMedium"
$s$:Sc/DoseAtPhantom2/Component = "Phantom2"
$b$:Sc/DoseAtPhantom2/OutputToConsole = "TRUE"
$s$:Sc/DoseAtPhantom2/OutputFile = "DoseAtPhantom2"
$s$:Sc/DoseAtPhantom2/OutputType = "csv"
$b$:Sc/DoseAtPhantom2/Visualize = "TRUE"

$s$:Sc/EDepAtPhantom2/Quantity = "EnergyDeposit"
$s$:Sc/EDepAtPhantom2/Component = "Phantom2"

$s$:Gr/ViewA/Type = "OpenGL"
$i$:Gr/ViewA/WindowSizeX = 900
$i$:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
g:Gr/ViewA/Projection = "Perspective"
g:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
$b$:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

$s$:So/Example/Type = "Beam"
$s$:So/Example/Component = "BeamPosition"
$s$:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
$u$:So/Example/BeamEnergySpread = 0.757504
$s$:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
Dose.txt

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"
s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "Air"
d:Ge/Phantom/HLX = 30.0 cm
d:Ge/Phantom/HLY = 30.0 cm
d:Ge/Phantom/HLZ = 10.0 cm
d:Ge/Phantom/TransZ = -30.0 cm
g:Ge/Phantom/Color = "blue"
s:Sc/DoseAtPhantom/Quantity = "DoseToMedium"
s:Sc/DoseAtPhantom/Component = "Phantom"
b:Sc/DoseAtPhantom/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"

#s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
g:So/Example/BeamPositionDistribution = "Gaussian"
g:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm

d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

b:Ts/PauseBeforeQuit = "True"
### DoseInVoxelMaterials.txt

- **Ge/World/Material** = "Vacuum"
- **Ge/World/HLX** = 2.0 m
- **Ge/World/HLY** = 2.0 m
- **Ge/World/HLZ** = 2.0 m
- **Ge/World/Invisible** = "True"
- **Ge/Phantom/Type** = "TsBox"
- **Ge/Phantom/Parent** = "World"
- **Ge/Phantom/Material** = "G4_WATER"
- **Ge/Phantom/HLX** = 30.0 cm
- **Ge/Phantom/HLY** = 30.0 cm
- **Ge/Phantom/HLZ** = 30.0 cm
- **Ge/Phantom/TransZ** = -30.0 cm
- **Ge/Phantom/Color** = "blue"
- **Ge/Phantom/XBins** = 10
- **Ge/Phantom/YBins** = 10
- **Ge/Phantom/ZBins** = 10
- **includeFile** = VoxelMaterials.txt
- **Ge/Phantom/RotY** = 270 deg
- **Sc/DoseAtPhantom/Quantity** = "DoseToMedium"
- **Sc/DoseAtPhantom/Component** = "Phantom"
- **Sc/DoseAtPhantom/OutputToConsole** = "TRUE"
- **Sc/DoseAtPhantom/IfOutputFileAlreadyExists** = "Overwrite"
- **Gr/ViewA/Type** = "OpenGL"
- **Gr/ViewA/WindowSizeX** = 900
- **Gr/ViewA/WindowSizeY** = 900
- **Gr/ViewA/Theta** = 55 deg
- **Gr/ViewA/Phi** = 20 deg
- **Gr/ViewA/Projection** = "Perspective"
- **Gr/ViewA/PerspectiveAngle** = 30 deg
- **Gr/ViewA/Zoom** = 1.3
- **Gr/ViewA/HiddenLineRemovalForTrajectories** = "True"
- **Ph/Default/Modules** = 1 "g4em-standard_opt0"
- **So/Example/Type** = "Beam"
DoseToMediumVsWater.txt

# Demonstrates dose to medium versus dose to water.

a:Ge/World/Material = "Vacuum"
b:Ge/World/Invisible = "True"
s:Ge/WaterBox/Type = "TsBox"
s:Ge/WaterBox/Parent = "World"
s:Ge/WaterBox/Material = "Gd_WATER"
d:Ge/WaterBox/HLX = 10.0 cm
d:Ge/WaterBox/HLY = 10.0 cm
d:Ge/WaterBox/HLZ = 2.0 cm
d:Ge/WaterBox/TransX = 0. cm
d:Ge/WaterBox/TransY = 0. cm
d:Ge/WaterBox/TransZ = 10. cm
d:Ge/WaterBox/RotX = 0. deg
d:Ge/WaterBox/RotY = 0. deg
d:Ge/WaterBox/RotZ = 0. deg
s:Ge/WaterBox/Color = "blue"
as:Ge/LeadBox/Type = "TsBox"
as:Ge/LeadBox/Parent = "World"
as:Ge/LeadBox/Material = "Lead"
ad:Ge/LeadBox/HLX = 10.0 cm
ad:Ge/LeadBox/HLY = 10.0 cm
ad:Ge/LeadBox/HLZ = 5.0 cm
d:Ge/LeadBox/TransX = 0. cm
d:Ge/LeadBox/TransY = 0. cm
d:Ge/LeadBox/TransZ = 0. cm
d:Ge/LeadBox/RotX = 0. deg
d:Ge/LeadBox/RotY = 0. deg
d:Ge/LeadBox/RotZ = 0. deg
s:Ge/LeadBox/Color = "blue"

s:Sc/1WaterBoxDoseToMedium/Quantity = "DoseToMedium"
s:Sc/1WaterBoxDoseToMedium/Component = "WaterBox"
b:Sc/1WaterBoxDoseToMedium/OutputToConsole = "TRUE"
s:Sc/1WaterBoxDoseToMedium/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/2WaterBoxDoseToWater/Quantity = "DoseToWater"
s:Sc/2WaterBoxDoseToWater/Component = "WaterBox"
b:Sc/2WaterBoxDoseToWater/OutputToConsole = "TRUE"
s:Sc/2WaterBoxDoseToWater/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/3WaterBoxDoseToMaterialWater/Quantity = "DoseToMaterial"
s:Sc/3WaterBoxDoseToMaterialWater/Material = "G4_WATER"
s:Sc/3WaterBoxDoseToMaterialWater/Component = "WaterBox"
b:Sc/3WaterBoxDoseToMaterialWater/OutputToConsole = "TRUE"
s:Sc/3WaterBoxDoseToMaterialWater/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/4WaterBoxDoseToMaterialLead/Quantity = "DoseToMaterial"
s:Sc/4WaterBoxDoseToMaterialLead/Material = "Lead"
s:Sc/4WaterBoxDoseToMaterialLead/Component = "WaterBox"
b:Sc/4WaterBoxDoseToMaterialLead/OutputToConsole = "TRUE"
s:Sc/4WaterBoxDoseToMaterialLead/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/5LeadBoxDoseToMedium/Quantity = "DoseToMedium"
s:Sc/5LeadBoxDoseToMedium/Component = "LeadBox"
b:Sc/5LeadBoxDoseToMedium/OutputToConsole = "TRUE"
s:Sc/5LeadBoxDoseToMedium/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/6LeadBoxDoseToWater/Quantity = "DoseToWater"
s:Sc/6LeadBoxDoseToWater/Component = "LeadBox"
b:Sc/6LeadBoxDoseToWater/OutputToConsole = "TRUE"
s:Sc/6LeadBoxDoseToWater/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/7LeadBoxDoseToMaterialWater/Quantity = "DoseToMaterial"
s:Sc/7LeadBoxDoseToMaterialWater/Material = "G4_WATER"
s:Sc/7LeadBoxDoseToMaterialWater/Component = "LeadBox"
b:Sc/7LeadBoxDoseToMaterialWater/OutputToConsole = "TRUE"
s:Sc/7LeadBoxDoseToMaterialWater/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/8LeadBoxDoseToMaterialLead/Quantity = "DoseToMaterial"
s:Sc/8LeadBoxDoseToMaterialLead/Material = "Lead"
s:Sc/8LeadBoxDoseToMaterialLead/Component = "LeadBox"
b:Sc/8LeadBoxDoseToMaterialLead/OutputToConsole = "TRUE"
s:Sc/8LeadBoxDoseToMaterialLead/IfOutputFileAlreadyExists = "Overwrite"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "gamma"
s:So/Example/BeamEnergy = 169.23 MeV
w:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
s:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
DoseVolumeHistogram.txt

// Score dose to a cumulative volume histogram

Ge/World/Material = "Vacuum"
Ge/World/HLX = 2.0 m
Ge/World/HLY = 2.0 m
Ge/World/HLZ = 2.0 m
Ge/Phantom/Type = "TsBox"
Ge/Phantom/Parent = "World"
Ge/Phantom/Material = "Lead"
Ge/Phantom/HLX = 30.0 cm
Ge/Phantom/HLY = 30.0 cm
Ge/Phantom/HLZ = 10.0 cm
Ge/Phantom/TransX = 0. cm
Ge/Phantom/TransY = 0. cm
Ge/Phantom/TransZ = -30. cm
Ge/Phantom/RotX = 0. deg
Ge/Phantom/RotY = 0. deg
Ge/Phantom/RotZ = 0. deg
Ge/Phantom/Color = "blue"
Ge/Phantom/XBins = 10
Ge/Phantom/YBins = 10
Ge/Phantom/ZBins = 10
Sc/DoseAtPhantom/Quantity = "DoseToMedium"
Sc/DoseAtPhantom/Component = "Phantom"
b:Sc/DoseAtPhantom/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/DoseAtPhantom/Report = 2 "Sum" "CumulativeVolumeHistogram"
i:Sc/DoseAtPhantom/HistogramBins = 100
d:Sc/DoseAtPhantom/HistogramMin = 0. Gy
d:Sc/DoseAtPhantom/HistogramMax = .0001 Gy
Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
d:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
# Demonstrate scoring energy and dose in a simple water phantom

```plaintext
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "Lead"
d:Ge/Phantom/HLX = 30.0 cm
d:Ge/Phantom/HLY = 30.0 cm
d:Ge/Phantom/HLZ = 10.0 cm
d:Ge/Phantom/TransZ = -30.0 cm
s:Ge/Phantom/Color = "blue"
```

```plaintext
s:Sc/EnergyDepAtPhantom/Quantity = "EnergyDeposit"
s:Sc/EnergyDepAtPhantom/Component = "Phantom"
b:Sc/EnergyDepAtPhantom/OutputToConsole = "TRUE"
s:Sc/EnergyDepAtPhantom/IfOutputFileAlreadyExists = "Overwrite"

s:Sc/DoseAtPhantom/Quantity = "DoseToMedium"
s:Sc/DoseAtPhantom/Component = "Phantom"
b:Sc/DoseAtPhantom/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
```

```plaintext
s:Gr/ViewA/Type = "OpenGL"
b:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
```
d:Gr/ViewA/Theta  =  55 deg
d:Gr/ViewA/Phi   =  20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom   = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

a:So/Example/Type    =  "Beam"
a:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

b:Ts/PauseBeforeQuit = "True"

EnergyDivisions.txt

# Score surface track count of primary protons
# in energy bins in two phantoms
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX    = 2.0 m
d:Ge/World/HLY    = 2.0 m
d:Ge/World/HLZ    = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/WaterBox/Type    = "TsBox"
s:Ge/WaterBox/Parent  = "World"
s:Ge/WaterBox/Material = "G4_WATER"
d:Ge/WaterBox/HLX    = 30.0 cm
d:Ge/WaterBox/HLY    = 30.0 cm
d:Ge/WaterBox/HLZ    = 2.0 cm
d:Ge/WaterBox/TransZ = -20. cm

s:Ge/LeadBox/Type    = "TsBox"
s:Ge/LeadBox/Parent  = "World"
s:Ge/LeadBox/Material = "Lead"
d:Ge/LeadBox/HLX    = 30.0 cm
d:Ge/LeadBox/HLY    = 30.0 cm
d:Ge/LeadBox/HLZ    = 10.0 cm
d:Ge/LeadBox/TransZ = -40. cm
d:Ge/LeadBox/RotX   = 0. deg
d:Ge/LeadBox/RotY   = 0. deg
192 Chapter 20. Scoring
EnergyInBinnedCylinder.txt

# Demonstrates binning by R, Phi and Z in a cylinder

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 1.0 m
d:Ge/World/HLY = 1.0 m
d:Ge/World/HLZ = 1.0 m
d:Ge/World/MaxStepSize = 1.0 m

s:Ge/ScoringCylinder/Type = "TsCylinder"
s:Ge/ScoringCylinder/Parent = "World"
s:Ge/ScoringCylinder/Material = "G4_WATER"
d:Ge/ScoringCylinder/RMin = 2. cm
d:Ge/ScoringCylinder/RMax = 40. cm
d:Ge/ScoringCylinder/HL = 40. cm
d:Ge/ScoringCylinder/SPhi = 0. deg
d:Ge/ScoringCylinder/DPhi = 360. deg
i:Ge/ScoringCylinder/RBins = 6
i:Ge/ScoringCylinder/PhiBins = 6
i:Ge/ScoringCylinder/ZBins = 6

s:Sc/MyScorer/Quantity = "EnergyDeposit"
s:Sc/MyScorer/Component = "ScoringCylinder"
s:Sc/MyScorer/OutputFile = "Energy_Test"
s:Sc/MyScorer/OutputType = "csv"
b:Sc/MyScorer/OutputToConsole = "False"
s:Sc/MyScorer/IfOutputFileAlreadyExists = "Overwrite"

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 2.
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 1000

i:Ts/ShowHistoryCountAtInterval = 100
Filters.txt

`# Test every kind of scoring filter`

s:Ge/World/Material = "Vacuum"
s:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/Foil/Type = "TsBox"
s:Ge/Foil/Parent = "World"
s:Ge/Foil/Material = "Lead"
d:Ge/Foil/HLX = 30.0 cm
d:Ge/Foil/HLY = 30.0 cm
d:Ge/Foil/HLZ = .5 cm
d:Ge/Foil/TransX = 0. cm
d:Ge/Foil/TransY = 0. cm
d:Ge/Foil/TransZ = 0. cm
d:Ge/Foil/RotX = 0. deg
d:Ge/Foil/RotY = 0. deg
d:Ge/Foil/RotZ = 0. deg

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 20.0 cm
d:Ge/Box/HLY = 20.0 cm
d:Ge/Box/HLZ = 20.0 cm
d:Ge/Box/TransX = 0. cm
d:Ge/Box/TransY = 0. cm
d:Ge/Box/TransZ = -30. cm
d:Ge/Box/RotX = 0. deg
d:Ge/Box/RotY = 0. deg
d:Ge/Box/RotZ = 0. deg

s:Sc/OnlyIncludeParticlesOfGeneration/Quantity = "SurfaceTrackCount"
bs:Sc/OnlyIncludeParticlesOfGeneration/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesOfGeneration/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeParticlesOfGeneration/OutputType = "csv"
b:Sc/OnlyIncludeParticlesOfGeneration/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/OnlyIncludeParticlesOfGeneration/OnlyIncludeParticlesOfGeneration = "Primary"
s:Sc/OnlyIncludeParticlesCharged/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesCharged/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesCharged/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeParticlesCharged/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesCharged/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/OnlyIncludeParticlesCharged/OnlyIncludeParticlesCharged = 1 "Positive"
s:Sc/OnlyIncludeParticlesNotCharged/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesNotCharged/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesNotCharged/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesNotCharged/OutputType = "csv"
s:Sc/OnlyIncludeParticlesNotCharged/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/OnlyIncludeParticlesNotCharged/OnlyIncludeParticlesNotCharged = 1 "Positive"

s:Sc/OnlyIncludeParticlesOfAtomicMass/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesOfAtomicMass/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesOfAtomicMass/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesOfAtomicMass/OutputType = "csv"
s:Sc/OnlyIncludeParticlesOfAtomicMass/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/OnlyIncludeParticlesOfAtomicMass/OnlyIncludeParticlesOfAtomicMass = 1

s:Sc/OnlyIncludeParticlesNotOfAtomicMass/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesNotOfAtomicMass/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesNotOfAtomicMass/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesNotOfAtomicMass/OutputType = "csv"
s:Sc/OnlyIncludeParticlesNotOfAtomicMass/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/OnlyIncludeParticlesNotOfAtomicMass/OnlyIncludeParticlesNotOfAtomicMass = 1

s:Sc/OnlyIncludeParticlesOfAtomicNumber/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesOfAtomicNumber/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesOfAtomicNumber/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesOfAtomicNumber/OutputType = "csv"
s:Sc/OnlyIncludeParticlesOfAtomicNumber/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/OnlyIncludeParticlesOfAtomicNumber/OnlyIncludeParticlesOfAtomicNumber = 1

s:Sc/OnlyIncludeParticlesNotOfAtomicNumber/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesNotOfAtomicNumber/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesNotOfAtomicNumber/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesNotOfAtomicNumber/OutputType = "csv"
s:Sc/OnlyIncludeParticlesNotOfAtomicNumber/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/OnlyIncludeParticlesNotOfAtomicNumber/OnlyIncludeParticlesNotOfAtomicNumber = 1

s:Sc/OnlyIncludeParticlesWithInitialKEBelow/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesWithInitialKEBelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKEBelow/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesWithInitialKEBelow/OutputType = "csv"
s:Sc/OnlyIncludeParticlesWithInitialKEBelow/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeParticlesWithInitialKEBelow/OnlyIncludeParticlesWithInitialKEBelow = \(-100.\) MeV

s:Sc/OnlyIncludeParticlesWithInitialKENotBelow/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesWithInitialKENotBelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKENotBelow/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesWithInitialKENotBelow/OutputType = "csv"
s:Sc/OnlyIncludeParticlesWithInitialKENotBelow/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeParticlesWithInitialKENotBelow/OnlyIncludeParticlesWithInitialKENotBelow = 100. MeV

s:Sc/OnlyIncludeParticlesWithInitialKE/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesWithInitialKE/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKE/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesWithInitialKE/OutputType = "csv"
s:Sc/OnlyIncludeParticlesWithInitialKE/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeParticlesWithInitialKE/OnlyIncludeParticlesWithInitialKE = 100. MeV

s:Sc/OnlyIncludeParticlesWithInitialKENot/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesWithInitialKENot/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKENot/OutputToConsole = "TRUE"
TOPAS Documentation, Release 3.0

Chapter 20. Scoring

```plaintext
Sc/OnlyIncludeParticlesWithInitialKENot/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialKENot/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialKENot/OnlyIncludeParticlesWithInitialKENot = 100. MeV

Sc/OnlyIncludeParticlesWithInitialKEAbove/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialKEAbove/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKEAbove/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialKEAbove/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialKEAbove/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialKEAbove/OnlyIncludeParticlesWithInitialKEAbove = 100. MeV

Sc/OnlyIncludeParticlesWithInitialKENotAbove/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialKENotAbove/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialKENotAbove/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialKENotAbove/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialKENotAbove/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialKENotAbove/OnlyIncludeParticlesWithInitialKENotAbove = 100. MeV

Sc/OnlyIncludeParticlesWithInitialMomentumBelow/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialMomentumBelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialMomentumBelow/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialMomentumBelow/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialMomentumBelow/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialMomentumBelow/OnlyIncludeParticlesWithInitialMomentumBelow = 100. MeV

Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialMomentumNotBelow/OnlyIncludeParticlesWithInitialMomentumNotBelow = 100. MeV

Sc/OnlyIncludeParticlesWithInitialMomentum/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialMomentum/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialMomentum/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialMomentum/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialMomentum/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialMomentum/OnlyIncludeParticlesWithInitialMomentum = 100. MeV

Sc/OnlyIncludeParticlesWithInitialMomentumNot/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialMomentumNot/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialMomentumNot/OutputToConsole = "TRUE"
Sc/OnlyIncludeParticlesWithInitialMomentumNot/OutputType = "csv"
Sc/OnlyIncludeParticlesWithInitialMomentumNot/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeParticlesWithInitialMomentumNot/OnlyIncludeParticlesWithInitialMomentumNot = 100. MeV

Sc/OnlyIncludeParticlesWithInitialMomentumAbove/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeParticlesWithInitialMomentumAbove/Surface = "Box/ZPlusSurface"
```
b:Sc/OnlyIncludeParticlesWithInitialMomentumAbove/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeParticlesWithInitialMomentumAbove/OutputType = "csv"
b:Sc/OnlyIncludeParticlesWithInitialMomentumAbove/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeParticlesWithInitialMomentumAbove/OnlyIncludeParticlesWithInitialMomentumAbove = 100. MeV

d:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/OutputType = "csv"
b:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeParticlesWithInitialMomentumNotAbove/OnlyIncludeParticlesWithInitialMomentumNotAbove = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKEbelow/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKEbelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKEbelow/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKEbelow/OutputType = "csv"
b:Sc/OnlyIncludeIfIncidentParticleKEbelow/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeIfIncidentParticleKEbelow/OnlyIncludeIfIncidentParticleKEbelow = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKENotBelow/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKENotBelow/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKENotBelow/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKENotBelow/OutputType = "csv"
b:Sc/OnlyIncludeIfIncidentParticleKENotBelow/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeIfIncidentParticleKENotBelow/OnlyIncludeIfIncidentParticleKENotBelow = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKE/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKE/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKE/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKE/OutputType = "csv"
b:Sc/OnlyIncludeIfIncidentParticleKE/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeIfIncidentParticleKE/OnlyIncludeIfIncidentParticleKE = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKENot/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKENot/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKENot/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKENot/OutputType = "csv"
b:Sc/OnlyIncludeIfIncidentParticleKENot/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeIfIncidentParticleKENot/OnlyIncludeIfIncidentParticleKENot = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKEAbove/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKEAbove/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKEAbove/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKEAbove/OutputType = "csv"
b:Sc/OnlyIncludeIfIncidentParticleKEAbove/IfOutputFileAlreadyExists = "Overwrite"
d:Sc/OnlyIncludeIfIncidentParticleKEAbove/OnlyIncludeIfIncidentParticleKEAbove = 100. MeV

d:Sc/OnlyIncludeIfIncidentParticleKENotAbove/Quantity = "SurfaceTrackCount"
d:Sc/OnlyIncludeIfIncidentParticleKENotAbove/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfIncidentParticleKENotAbove/OutputToConsole = "TRUE"
b:Sc/OnlyIncludeIfIncidentParticleKENotAbove/OutputType = "csv"
OnlyIncludeIfIncidentParticleKENotAbove/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleKENotAbove = 100. MeV

OnlyIncludeIfIncidentParticleMomentumBelow/Quantity = "SurfaceTrackCount"
OnlyIncludeIfIncidentParticleMomentumBelow/Surface = "Box/ZPlusSurface"
OnlyIncludeIfIncidentParticleMomentumBelow/OutputToConsole = "TRUE"
OnlyIncludeIfIncidentParticleMomentumBelow/OutputType = "csv"
OnlyIncludeIfIncidentParticleMomentumBelow/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleMomentumBelow = 100. MeV

OnlyIncludeIfIncidentParticleMomentumNotBelow/Quantity = "SurfaceTrackCount"
OnlyIncludeIfIncidentParticleMomentumNotBelow/Surface = "Box/ZPlusSurface"
OnlyIncludeIfIncidentParticleMomentumNotBelow/OutputToConsole = "TRUE"
OnlyIncludeIfIncidentParticleMomentumNotBelow/OutputType = "csv"
OnlyIncludeIfIncidentParticleMomentumNotBelow/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleMomentumNotBelow = 100. MeV

OnlyIncludeIfIncidentParticleMomentumNot/Quantity = "SurfaceTrackCount"
OnlyIncludeIfIncidentParticleMomentumNot/Surface = "Box/ZPlusSurface"
OnlyIncludeIfIncidentParticleMomentumNot/OutputToConsole = "TRUE"
OnlyIncludeIfIncidentParticleMomentumNot/OutputType = "csv"
OnlyIncludeIfIncidentParticleMomentumNot/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleMomentumNot = 100. MeV

OnlyIncludeIfIncidentParticleMomentumAbove/Quantity = "SurfaceTrackCount"
OnlyIncludeIfIncidentParticleMomentumAbove/Surface = "Box/ZPlusSurface"
OnlyIncludeIfIncidentParticleMomentumAbove/OutputToConsole = "TRUE"
OnlyIncludeIfIncidentParticleMomentumAbove/OutputType = "csv"
OnlyIncludeIfIncidentParticleMomentumAbove/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleMomentumAbove = 100. MeV

OnlyIncludeIfIncidentParticleMomentumNotAbove/Quantity = "SurfaceTrackCount"
OnlyIncludeIfIncidentParticleMomentumNotAbove/Surface = "Box/ZPlusSurface"
OnlyIncludeIfIncidentParticleMomentumNotAbove/OutputToConsole = "TRUE"
OnlyIncludeIfIncidentParticleMomentumNotAbove/OutputType = "csv"
OnlyIncludeIfIncidentParticleMomentumNotAbove/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfIncidentParticleMomentumNotAbove = 100. MeV

OnlyIncludeParticlesFromProcess/Quantity = "SurfaceTrackCount"
OnlyIncludeParticlesFromProcess/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesFromProcess/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesFromProcess/OutputType = "csv"
s:Sc/OnlyIncludeParticlesFromProcess/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesFromProcess/OnlyIncludeParticlesFromProcess = 1 "hIoni"

s:Sc/OnlyIncludeParticlesNotFromProcess/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesNotFromProcess/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesNotFromProcess/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesNotFromProcess/OutputType = "csv"
s:Sc/OnlyIncludeParticlesNotFromProcess/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesNotFromProcess/OnlyIncludeParticlesNotFromProcess = 1 "hIoni"

s:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/OutputType = "csv"
s:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorFromProcess/OnlyIncludeIfParticleOrAncestorFromProcess = 1 "hIoni"

s:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/OutputType = "csv"
s:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorNotFromProcess/OnlyIncludeIfParticleOrAncestorNotFromProcess = 1 "proton"

s:Sc/OnlyIncludeParticlesNamed/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeParticlesNamed/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeParticlesNamed/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeParticlesNamed/OutputType = "csv"
s:Sc/OnlyIncludeParticlesNamed/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesNamed/OnlyIncludeParticlesNamed = 1 "proton"

s:Sc/OnlyIncludeIfParticleOrAncestorNamed/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeIfParticleOrAncestorNamed/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorNamed/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeIfParticleOrAncestorNamed/OutputType = "csv"
s:Sc/OnlyIncludeIfParticleOrAncestorNamed/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorNamed/OnlyIncludeIfParticleOrAncestorNamed = 1 "proton"

s:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/Quantity = "SurfaceTrackCount"
s:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/OutputToConsole = "TRUE"
s:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/OutputType = "csv"
s:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorNotNamed/OnlyIncludeIfParticleOrAncestorNotNamed => 1 "proton"

sv:Sc/OnlyIncludeParticlesFromVolume/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesFromVolume/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesFromVolume/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesFromVolume/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesFromVolume/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesFromVolume/OnlyIncludeParticlesFromVolume = 2 "World" "Foil"

sv:Sc/OnlyIncludeParticlesNotFromVolume/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesNotFromVolume/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesNotFromVolume/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesNotFromVolume/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesNotFromVolume/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesNotFromVolume/OnlyIncludeParticlesNotFromVolume = 2 "World" "Foil"

sv:Sc/OnlyIncludeParticlesFromComponent/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesFromComponent/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesFromComponent/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesFromComponent/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesFromComponent/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesFromComponent/OnlyIncludeParticlesFromComponent = 2 "World" "Foil"

sv:Sc/OnlyIncludeParticlesNotFromComponent/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesNotFromComponent/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesNotFromComponent/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesNotFromComponent/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesNotFromComponent/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesNotFromComponent/OnlyIncludeParticlesNotFromComponent = 2 "World" "Foil"

sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesFromComponentOrSubComponentsOf/OnlyIncludeParticlesFromComponentOrSubComponentsOf = 2 "World" "Foil"

sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/OutputType = "csv"
sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf/OnlyIncludeParticlesNotFromComponentOrSubComponentsOf = 2 "World" "Foil"

sv:Sc/OnlyIncludeIfParticleOrAncestorFromVolume/Quantity = "SurfaceTrackCount"
sv:Sc/OnlyIncludeIfParticleOrAncestorFromVolume/Surface = "Box/ZPlusSurface"
sv:Sc/OnlyIncludeIfParticleOrAncestorFromVolume/OutputToConsole = "TRUE"
sv:Sc/OnlyIncludeIfParticleOrAncestorFromVolume/OutputType = "csv"
OnlyIncludeIfParticleOrAncestorFromVolume/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorFromVolume = 2 "World" "Foil"

OnlyIncludeIfParticleOrAncestorNotFromVolume/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleOrAncestorNotFromVolume/Surface = "Box/ZPlusSurface"

OnlyIncludeIfParticleOrAncestorNotFromVolume/OutputToConsole = "TRUE"

OnlyIncludeIfParticleOrAncestorNotFromVolume/OutputType = "csv"

OnlyIncludeIfParticleOrAncestorNotFromVolume/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorNotFromVolume = 2 "World" "Foil"

OnlyIncludeIfParticleOrAncestorFromComponent/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleOrAncestorFromComponent/Surface = "Box/ZPlusSurface"

OnlyIncludeIfParticleOrAncestorFromComponent/OutputToConsole = "TRUE"

OnlyIncludeIfParticleOrAncestorFromComponent/OutputType = "csv"

OnlyIncludeIfParticleOrAncestorFromComponent/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorFromComponent = 2 "World" "Foil"

OnlyIncludeIfParticleOrAncestorNotFromComponent/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleOrAncestorNotFromComponent/Surface = "Box/ZPlusSurface"

OnlyIncludeIfParticleOrAncestorNotFromComponent/OutputToConsole = "TRUE"

OnlyIncludeIfParticleOrAncestorNotFromComponent/OutputType = "csv"

OnlyIncludeIfParticleOrAncestorNotFromComponent/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorNotFromComponent = 2 "World" "Foil"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf/OutputToConsole = "TRUE"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf/OutputType = "csv"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorFromComponentOrSubComponentsOf = 2 "World" "Foil"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf/OutputToConsole = "TRUE"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf/OutputType = "csv"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"

OnlyIncludeIfParticleOrAncestorNotFromComponentOrSubComponentsOf = 2 "World" "Foil"

OnlyIncludeIfParticleInteractedInVolume/Quantity = "SurfaceTrackCount"

OnlyIncludeIfParticleInteractedInVolume/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleInteractedInVolume/OutputToConsole = "TRUE"
OnlyIncludeIfParticleInteractedInVolume/OutputType = "csv"
OnlyIncludeIfParticleInteractedInVolume/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleInteractedInVolume/OnlyIncludeIfParticleInteractedInVolume = 2 "World" "Foil"
OnlyIncludeIfParticleNotInteractedInVolume/Quantity = "SurfaceTrackCount"
OnlyIncludeIfParticleNotInteractedInVolume/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleNotInteractedInVolume/OutputToConsole = "TRUE"
OnlyIncludeIfParticleNotInteractedInVolume/OutputType = "csv"
OnlyIncludeIfParticleNotInteractedInVolume/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleNotInteractedInVolume/OnlyIncludeIfParticleNotInteractedInVolume = 2 "World" "Foil"
OnlyIncludeIfParticleInteractedInComponent/Quantity = "SurfaceTrackCount"
OnlyIncludeIfParticleInteractedInComponent/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleInteractedInComponent/OutputToConsole = "TRUE"
OnlyIncludeIfParticleInteractedInComponent/OutputType = "csv"
OnlyIncludeIfParticleInteractedInComponent/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleInteractedInComponent/OnlyIncludeIfParticleInteractedInComponent = 2 "World" "Foil"
OnlyIncludeIfParticleNotInteractedInComponent/Quantity = "SurfaceTrackCount"
OnlyIncludeIfParticleNotInteractedInComponent/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleNotInteractedInComponent/OutputToConsole = "TRUE"
OnlyIncludeIfParticleNotInteractedInComponent/OutputType = "csv"
OnlyIncludeIfParticleNotInteractedInComponent/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleNotInteractedInComponent/OnlyIncludeIfParticleNotInteractedInComponent = 2 "World" "Foil"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/OutputToConsole = "TRUE"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/OutputType = "csv"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf/OnlyIncludeIfParticleInteractedInComponentOrSubComponentsOf = 2 "World" "Foil"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/OutputToConsole = "TRUE"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/OutputType = "csv"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf/OnlyIncludeIfParticleNotInteractedInComponentOrSubComponentsOf = 2 "World" "Foil"
OnlyIncludeIfParticleOrAncestorInteractedInVolume/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInVolume/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInVolume/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInVolume/OutputType = "csv"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInVolume/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInVolume = 2 "World" "Foil"

Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume/OutputType = "csv"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInVolume = 2 "World" "Foil"

Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent/OutputType = "csv"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponent = 2 "World" "Foil"

Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent/OutputType = "csv"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponent = 2 "World" "Foil"

Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf/OutputType = "csv"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleOrAncestorInteractedInComponentOrSubComponentsOf = 2 "World" "Foil"

Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/OutputToConsole = "TRUE"
`Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/OutputType = "csv"`
`Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
`Sc/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf/OnlyIncludeIfParticleOrAncestorNotInteractedInComponentOrSubComponentsOf = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleTraversedVolume/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleTraversedVolume/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleTraversedVolume/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleTraversedVolume/OutputType = "csv"
Sc/OnlyIncludeIfParticleTraversedVolume/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleTraversedVolume/OnlyIncludeIfParticleTraversedVolume = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleNotTraversedVolume/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleNotTraversedVolume/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleNotTraversedVolume/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleNotTraversedVolume/OutputType = "csv"
Sc/OnlyIncludeIfParticleNotTraversedVolume/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleNotTraversedVolume/OnlyIncludeIfParticleNotTraversedVolume = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleTraversedComponent/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleTraversedComponent/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleTraversedComponent/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleTraversedComponent/OutputType = "csv"
Sc/OnlyIncludeIfParticleTraversedComponent/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleTraversedComponent/OnlyIncludeIfParticleTraversedComponent = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleNotTraversedComponent/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleNotTraversedComponent/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleNotTraversedComponent/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleNotTraversedComponent/OutputType = "csv"
Sc/OnlyIncludeIfParticleNotTraversedComponent/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleNotTraversedComponent/OnlyIncludeIfParticleNotTraversedComponent = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/OutputToConsole = "TRUE"
Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/OutputType = "csv"
Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
Sc/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf/OnlyIncludeIfParticleTraversedComponentOrSubComponentsOf = 2 "World" "Foil"

`Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/Quantity = "SurfaceTrackCount"
Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/Surface = "Box/ZPlusSurface"
Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/OutputToConsole = "TRUE"`
s:Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/OutputType = "csv"
s:Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf/
  OnlyIncludeIfParticleNotTraversedComponentOrSubComponentsOf = 2 "World" "Foil"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/Quantity = "SurfaceTrackCount"
bs:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/OutputToConsole = "TRUE"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/OutputType = "csv"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorTraversedVolume/
  OnlyIncludeIfParticleOrAncestorTraversedVolume = 2 "World" "Foil"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/Quantity = "SurfaceTrackCount"
bs:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/OutputToConsole = "TRUE"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/OutputType = "csv"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedVolume/
  OnlyIncludeIfParticleOrAncestorNotTraversedVolume = 2 "World" "Foil"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/Quantity = "SurfaceTrackCount"
bs:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/OutputToConsole = "TRUE"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/OutputType = "csv"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponent/
  OnlyIncludeIfParticleOrAncestorTraversedComponent = 2 "World" "Foil"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/Quantity =
  "SurfaceTrackCount"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/Surface = "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/OutputToConsole = "TRUE"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/OutputType = "csv"
as:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorNotTraversedComponent/
  OnlyIncludeIfParticleOrAncestorNotTraversedComponent = 2 "World" "Foil"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/Quantity =
  "SurfaceTrackCount"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/Surface =
  "Box/ZPlusSurface"
b:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/
  OutputToConsole = "TRUE"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/
  OutputType = "csv"
as:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf/
  OnlyIncludeIfParticleOrAncestorTraversedComponentOrSubComponentsOf = 2 "World" "Foil"
# Test scoring in geometry of particle passing through a foil to a box

```plaintext
Ge/World/Material = "Vacuum"
```

FoilToBox.txt

---

206 Chapter 20. Scoring
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"

s:Ge/Foil/Type = "TsBox"
s:Ge/Foil/Parent = "World"
s:Ge/Foil/Material = "Lead"
d:Ge/Foil/HLX = 30.0 cm
d:Ge/Foil/HLY = 30.0 cm
d:Ge/Foil/HLZ = .5 cm
d:Ge/Foil/TransX = 0. cm
d:Ge/Foil/TransY = 0. cm
d:Ge/Foil/TransZ = 0. cm
d:Ge/Foil/RotX = 0. deg
d:Ge/Foil/RotY = 0. deg
d:Ge/Foil/RotZ = 0. deg
i:Ge/Foil/XBins = 1
i:Ge/Foil/YBins = 1
i:Ge/Foil/ZBins = 1

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 20.0 cm
d:Ge/Box/HLY = 20.0 cm
d:Ge/Box/HLZ = 20.0 cm
d:Ge/Box/TransX = 0. cm
d:Ge/Box/TransY = 0. cm
d:Ge/Box/TransZ = -30. cm
d:Ge/Box/RotX = 0. deg
d:Ge/Box/RotY = 0. deg
d:Ge/Box/RotZ = 0. deg
s:Ge/Box/Color = "blue"
i:Ge/Box/ZBins = 2

s:Sc/MyScorer/Quantity = "SurfaceTrackCount"
s:Sc/MyScorer/Surface = "Box/ZPlusSurface"
b:Sc/MyScorer/OutputToConsole = "True"
s:Sc/MyScorer/OutputType = "csv"
s:Sc/MyScorer/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/MyScorer/EBins = 10
s:Sc/MyScorer/OnlyIncludeParticlesGoing = "Out"
sv:Sc/MyScorer/OnlyIncludeParticlesNamed = 1 "gamma"
s:Sc/MyScorer/OnlyIncludeParticlesOfGeneration = "secondary"
#d:Sc/MyScorer/EBinMin = 0. MeV
d:Sc/MyScorer/EBinMax = 110. MeV
#d:Sc/MyScorer/OnlyIncludeIfIncidentParticleKEAbove = 0 MeV

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
FoilToCylinder.txt

# Test scoring in geometry of particle passing through a foil to a cylinder

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"

d:Ge/Foil/Type = "TsBox"
s:Ge/Foil/Parent = "World"
s:Ge/Foil/Material = "Lead"
d:Ge/Foil/HLX = 30.0 cm
d:Ge/Foil/HLY = 30.0 cm
d:Ge/Foil/HLZ = .5 cm
d:Ge/Foil/TransX = 0. cm
d:Ge/Foil/TransY = 0. cm
d:Ge/Foil/TransZ = -10. cm
d:Ge/Foil/RotX = 0. deg
d:Ge/Foil/RotY = 0. deg
d:Ge/Foil/RotZ = 0. deg
s:Ge/Foil/Color = "blue"
i:Ge/Foil/XBins = 1
i:Ge/Foil/YBins = 1
i:Ge/Foil/ZBins = 1
s:Ge/Cylinder/Type = "TsCylinder"
s:Ge/Cylinder/Parent = "World"
s:Ge/Cylinder/Material = "Vacuum"
d:Ge/Cylinder/RMin = 50.0 cm
d:Ge/Cylinder/RMax = 55.0 cm
d:Ge/Cylinder/SPhi = 20.0 deg
d:Ge/Cylinder/DPhi = 100. deg
### Ge/Cylinder
- **Height (H)**: 50. cm
- **TransX**: 0. cm
- **TransY**: 0. cm
- **TransZ**: 30. cm
- **RotX**: 0. deg
- **RotY**: 0. deg
- **RotZ**: 0. deg
- **Color**: "blue"
- **RBins**: 2

### Sc/MyScorer
- **Quantity**: "SurfaceTrackCount"
- **Surface**: "Cylinder/OuterCurvedSurface"
- **OutputToConsole**: "true"
- **OutputType**: "csv"
- **IfOutputFileAlreadyExists**: "Overwrite"
- **EBins**: 10
- **OnlyIncludeParticlesGoing**: "Out"
- **OnlyIncludeParticlesNamed**: 1 "gamma"
- **OnlyIncludeParticlesOfGeneration**: "secondary"
- **EBinMin**: 0 MeV
- **EBinMax**: 110. MeV
- **IfOutputFileAlreadyExists**: "Overwrite"
- **OnlyIncludeIfIncidentParticleKEAbove**: 0 MeV

### Graphics
- **Type**: "OpenGL"
- **WindowSizeX**: 900
- **WindowSizeY**: 900
- **Theta**: 55 deg
- **Phi**: 20 deg
- **Projection**: "Perspective"
- **PerspectiveAngle**: 30 deg
- **Zoom**: 1.3
- **HiddenLineRemovalForTrajectories**: "true"

### So/Example
- **Type**: "Beam"
- **Component**: "BeamPosition"
- **BeamParticle**: "proton"
- **BeamEnergy**: 169.23 MeV
- **BeamEnergySpread**: 0.757504
- **BeamPositionDistribution**: "Gaussian"
- **BeamPositionCutoffShape**: "Ellipse"
- **BeamPositionCutoffX**: 10. cm
- **BeamPositionCutoffY**: 10. cm
- **BeamPositionSpreadX**: 0.65 cm
- **BeamPositionSpreadY**: 0.65 cm
- **BeamAngularDistribution**: "Gaussian"
- **BeamAngularCutoffX**: 90. deg
- **BeamAngularCutoffY**: 90. deg
- **BeamAngularSpreadX**: 0.0032 rad
- **BeamAngularSpreadY**: 0.0032 rad
- **NumberOfHistoriesInRun**: 10

### Ts/PauseBeforeQuit
- "true"
# Test scoring in geometry of particle passing through a foil to a sphere

```
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"
s:Ge/Foil/Type = "TsBox"
s:Ge/Foil/Parent = "World"
s:Ge/Foil/Material = "Lead"
d:Ge/Foil/HLX = 30.0 cm
d:Ge/Foil/HLY = 30.0 cm
d:Ge/Foil/HLZ = .5 cm
d:Ge/Foil/TransX = 0. cm
d:Ge/Foil/TransY = 0. cm
d:Ge/Foil/TransZ = -30. cm
d:Ge/Foil/RotX = 0. deg
d:Ge/Foil/RotY = 0. deg
d:Ge/Foil/RotZ = 0. deg
s:Ge/Foil/Color = "blue"
i:Ge/Foil/XBins = 1
i:Ge/Foil/YBins = 1
i:Ge/Foil/ZBins = 1

s:Ge/Sphere/Type = "TsSphere"
s:Ge/Sphere/Parent = "World"
s:Ge/Sphere/Material = "Vacuum"
d:Ge/Sphere/RMin = 50.0 cm
d:Ge/Sphere/RMax = 55.0 cm
d:Ge/Sphere/SPhi = 0.0 deg
d:Ge/Sphere/DPhi = 120. deg
d:Ge/Sphere/STheta = 0 deg
d:Ge/Sphere/DTheta = 180 deg
d:Ge/Sphere/TransX = 0. cm
d:Ge/Sphere/TransY = 0. cm
d:Ge/Sphere/TransZ = -30. cm
d:Ge/Sphere/RotX = 0. deg
d:Ge/Sphere/RotY = 0. deg
d:Ge/Sphere/RotZ = 0. deg
s:Ge/Sphere/Color = "blue"
i:Ge/Sphere/RBins = 2

s:Sc/MyScorer/Quantity = "SurfaceTrackCount"
s:Sc/MyScorer/Surface = "Sphere/InnerCurvedSurface"
b:Sc/MyScorer/OutputToConsole = "True"
s:Sc/MyScorer/OutputType = "csv"
b:i:Sc/MyScorer/IfOutputFileAlreadyExists = "Overwrite"
s:i:Sc/MyScorer/EBins = 10
s:Sc/MyScorer/OnlyIncludeParticlesGoing = "Out"
s:Sc/MyScorer/OnlyIncludeParticlesNamed = 1 "gamma"
s:Sc/MyScorer/OnlyIncludeParticlesOfGeneration = "secondary"
s:d:Sc/MyScorer/EBinMin = 0. MeV
j:d:Sc/MyScorer/EBinMax = 110. MeV
```

# Chapter 20. Scoring
Gated.txt

# Demonstrates Gated scoring.
# First scorer is gated, sometimes active, sometimes inactive,
# under the control of a time feature and outputs after each run.
# Second scorer is gated, sometimes active, sometimes inactive,
# under the control of a time feature and outputs only at end of session.
# Third scorer is always active and outputs only at end of session.

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
da:Sc/GatedAndOutputPerRun/Quantity = "Fluence"
da:Sc/GatedAndOutputPerRun/Component = "Box"
s:Sc/GatedAndOutputPerRun/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/GatedAndOutputPerRun/OutputToConsole = "True"
b:Sc/GatedAndOutputPerRun/Active = Tf/ScoringOnOff/Value
b:Sc/GatedAndOutputPerRun/OutputAfterRun = "True"
s:Sc/GatedAndOutputPerSession/Quantity = "Fluence"
# Demonstrates scoring by various geometry divisions.
# First scorer uses geometry of the regular world.
# Second scorer uses a parallel geometry with 2 X bins.
# Third scorer uses a parallel geometry with 2 Y bins.

```plaintext
W:Sc/GatedAndOutputPerSession/Component = "Box"
W:Sc/GatedAndOutputPerSession/IfOutputFileAlreadyExists = "Overwrite"
W:Sc/GatedAndOutputPerSession/OutputToConsole = "True"
W:Sc/GatedAndOutputPerSession/Active = Tf/ScoringOnOff/Value
W:Sc/NotGated/Quantity = "Fluence"
W:Sc/NotGated/Component = "Box"
W:Sc/NotGated/IfOutputFileAlreadyExists = "Overwrite"
W:Sc/NotGated/OutputToConsole = "True"
W:Gr/MyOGL/Type = "OpenGL"
W:Gr/MyOGL/WindowSizeX = 600
W:Gr/MyOGL/WindowSizeY = 600
W:Gr/MyOGL/WindowPosX = 0
W:Gr/MyOGL/WindowPosY = 0
W:Tf/ScoringOnOff/Function = "Step"
W:cl:Tf/ScoringOnOff/Times = 10 10 20 30 40 50 60 70 80 90 100 ms
W:cl:Tf/ScoringOnOff/Values = 10 "true" "false" "true" "false" "false" "true" "true" "false" "true" "false" "true"
W:Tf/Verbosity = 1
W:Tf/TimelineEnd = 100.0 ms
W:Tf/NumberOfSequentialTimes = 10
W:So/Example/Type = "Beam"
W:So/Example/Component = "BeamPosition"
W:So/Example/BeamParticle = "proton"
W:So/Example/BeamEnergy = 169.23 MeV
W:So/Example/BeamEnergySpread = 0.757504
W:So/Example/BeamPositionDistribution = "Gaussian"
W:So/Example/BeamPositionCutoffShape = "Ellipse"
W:So/Example/BeamPositionCutoffX = 10. cm
W:So/Example/BeamPositionCutoffY = 10. cm
W:So/Example/BeamPositionSpreadX = 0.65 cm
W:So/Example/BeamPositionSpreadY = 0.65 cm
W:So/Example/BeamAngularDistribution = "Gaussian"
W:So/Example/BeamAngularCutoffX = 90. deg
W:So/Example/BeamAngularCutoffY = 90. deg
W:So/Example/BeamAngularSpreadX = 0.0032 rad
W:So/Example/BeamAngularSpreadY = 0.0032 rad
W:So/Example/NumberOfHistoriesInRun = 10
W:Ts/PauseBeforeQuit = "True"
```
b:Ge/World/Invisible = "True"

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "G4_WATER"
d:Ge/Phantom/HLX = 30.0 cm
d:Ge/Phantom/HLY = 30.0 cm
d:Ge/Phantom/HLZ = 1.1 cm
d:Ge/Phantom/TransX = 0. cm
d:Ge/Phantom/TransY = 0. cm
d:Ge/Phantom/TransZ = -30. cm
d:Ge/Phantom/RotX = 0. deg
d:Ge/Phantom/RotY = 0. deg
d:Ge/Phantom/RotZ = 0. deg
s:Ge/Phantom/Color = "blue"

# Scores in regular world
s:Sc/DoseAtPhantom/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom/Component = "Phantom"
b:Sc/DoseAtPhantom/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"

# X binning causes creation of a parallel world for scoring
s:Sc/DoseAtPhantomParallel/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantomParallel/Component = "Phantom"
b:Sc/DoseAtPhantomParallel/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantomParallel/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/DoseAtPhantomParallel/XBins = 2

# Y binning causes creation of a parallel world for scoring
s:Sc/DoseAtPhantomParallel2/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantomParallel2/Component = "Phantom"
b:Sc/DoseAtPhantomParallel2/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantomParallel2/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/DoseAtPhantomParallel2/YBins = 2

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
v:Gr/ViewA/VisibleWorlds = 1 "All"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
s:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
Histograms.txt

# Create a variety of energy histograms

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "Lead"
d:Ge/Phantom/HLX = 30.0 cm
d:Ge/Phantom/HLY = 30.0 cm
d:Ge/Phantom/HLZ = 10.0 cm
d:Ge/Phantom/TransX = 0. cm
d:Ge/Phantom/TransY = 0. cm
d:Ge/Phantom/TransZ = -30. cm
d:Ge/Phantom/RotX = 0. deg
d:Ge/Phantom/RotY = 0. deg
d:Ge/Phantom/RotZ = 0. deg
s:Ge/Phantom/Color = "blue"

s:Sc/DoseAtPhantom/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom/Component = "Phantom"
b:Sc/DoseAtPhantom/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom/OutputType = "Root"
i:Sc/DoseAtPhantom/HistogramBins = 10
d:Sc/DoseAtPhantom/HistogramMin = 0. MeV
d:Sc/DoseAtPhantom/HistogramMax = 200. MeV

s:Sc/DoseAtPhantomByX/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantomByX/Component = "Phantom"
b:Sc/DoseAtPhantomByX/OutputToConsole = "TRUE"
s:Sc/DoseAtPhantomByX/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantomByX/OutputType = "Root"
i:Sc/DoseAtPhantomByX/XBins = 10
i:Sc/DoseAtPhantomByX/HistogramBins = 10
d:Sc/DoseAtPhantomByX/HistogramMin = 0. MeV
d:Sc/DoseAtPhantomByX/HistogramMax = 200. MeV

s:Sc/DoseAtPhantomByY/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantomByY/Component = "Phantom"
b:Sc/DoseAtPhantomByY/OutputToConsole = "TRUE"
# TOPAS Documentation, Release 3.0

## Sc/DoseAtPhantomByY
- IfOutputFileAlreadyExists = "Overwrite"
- OutputType = "Root"
- YBins = 10
- HistogramBins = 10
- HistogramMin = 0. MeV
- HistogramMax = 200. MeV

## Sc/DoseAtPhantomByZ
- Quantity = "EnergyDeposit"
- Component = "Phantom"
- OutputToConsole = "TRUE"
- IfOutputFileAlreadyExists = "Overwrite"
- OutputType = "Root"
- ZBins = 10
- HistogramBins = 10
- HistogramMin = 0. MeV
- HistogramMax = 200. MeV

## Sc/DoseAtPhantomByE
- Quantity = "EnergyDeposit"
- Component = "Phantom"
- OutputToConsole = "TRUE"
- IfOutputFileAlreadyExists = "Overwrite"
- OutputType = "Root"
- EBins = 10
- EBinMax = 200. MeV
- HistogramBins = 10
- HistogramMin = 0. MeV
- HistogramMax = 200. MeV

## Gr/ViewA
- Type = "OpenGL"
- WindowSizeX = 900
- WindowSizeY = 900
- Theta = 55 deg
- Phi = 20 deg
- Projection = "Perspective"
- PerspectiveAngle = 30 deg
- Zoom = 1.3
- HiddenLineRemovalForTrajectories = "True"

## So/Example
- Type = "Beam"
- Component = "BeamPosition"
- BeamParticle = "proton"
- BeamEnergy = 169.23 MeV
- BeamEnergySpread = 0.757504
- BeamPositionDistribution = "Gaussian"
- BeamPositionCutoffShape = "Ellipse"
- BeamPositionCutoffX = 10. cm
- BeamPositionCutoffY = 10. cm
- BeamPositionSpreadX = 0.65 cm
- BeamPositionSpreadY = 0.65 cm
- BeamAngularDistribution = "Gaussian"
- BeamAngularCutoffX = 90. deg
- BeamAngularCutoffY = 90. deg
- BeamAngularSpreadX = 0.0032 rad
- BeamAngularSpreadY = 0.0032 rad
- NumberOfHistoriesInRun = 10000
- ShowHistoryCountAtInterval = 100
Inactive.txt

``` lua
# Demonstrate effect of setting a scorer inactive
s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
s:Sc/WasActive/Quantity = "Fluence"
s:Sc/WasActive/Component = "Box"
s:Sc/WasActive/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/WasActive/OutputToConsole = "True"
s:Sc/WasInactive/Quantity = "Fluence"
s:Sc/WasInactive/Component = "Box"
s:Sc/WasInactive/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/WasInactive/OutputToConsole = "True"
b:Sc/WasInactive/Active = "False"
s:Gr/MyOGL/Type = "OpenGL"
i:Gr/MyOGL/WindowSizeX = 600
i:Gr/MyOGL/WindowSizeY = 600
i:Gr/MyOGL/WindowPosX = 0
i:Gr/MyOGL/WindowPosY = 0
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 100
i:Ts/ShowHistoryCountAtInterval = 10
b:Ts/PauseBeforeQuit = "True"
```

Ion.txt

``` lua
# Test filtering on ions
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
```
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/BeamPosition/Parent = "World"
s:Ge/BeamPosition/Type = "Group"
d:Ge/BeamPosition/TransX = 0.0 cm
d:Ge/BeamPosition/TransY = 0.0 cm
d:Ge/BeamPosition/TransZ = Ge/World/HLZ m
d:Ge/BeamPosition/RotX = 180. deg
d:Ge/BeamPosition/RotY = 0. deg
d:Ge/BeamPosition/RotZ = 0. deg

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "G4_WATER"
d:Ge/Phantom/HLX = 5.0 cm
d:Ge/Phantom/HLY = 5.0 cm
d:Ge/Phantom/HLZ = 5.0 cm
d:Ge/Phantom/TransX = 0. cm
d:Ge/Phantom/TransY = 0. cm
d:Ge/Phantom/TransZ = 0. cm
d:Ge/Phantom/RotX = 0. deg
d:Ge/Phantom/RotY = 0.0 deg
d:Ge/Phantom/RotZ = 0. deg
s:Ge/Phantom/Color = "blue"
i:Ge/Phantom/XBins = 1
i:Ge/Phantom/YBins = 1
i:Ge/Phantom/ZBins = 1

s:Sc/AllOfAtomicNumber6/Quantity = "EnergyDeposit"
s:Sc/AllOfAtomicNumber6/Component = "Phantom"
sv:Sc/AllOfAtomicNumber6/Report = 3 "Mean" "Count_In_Bin"
"Standard_Deviation"
b:Sc/AllOfAtomicNumber6/OutputToConsole = "True"
s:Sc/AllOfAtomicNumber6/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/AllOfAtomicNumber6/OnlyIncludeParticlesOfAtomicNumber = 6

s:Sc/IonsWithWildCards/Quantity = "EnergyDeposit"
s:Sc/IonsWithWildCards/Component = "Phantom"
sv:Sc/IonsWithWildCards/Report = 3 "Mean" "Count_In_Bin"
"Standard_Deviation"
b:Sc/IonsWithWildCards/OutputToConsole = "True"
s:Sc/IonsWithWildCards/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/IonsWithWildCards/OnlyIncludeParticlesNamed = 1 "GenericIon(6,*,*)"

s:Sc/IonsWithCharge3to5/Quantity = "EnergyDeposit"
s:Sc/IonsWithCharge3to5/Component = "Phantom"
sv:Sc/IonsWithCharge3to5/Report = 3 "Mean" "Count_In_Bin"
"Standard_Deviation"
b:Sc/IonsWithCharge3to5/OutputToConsole = "True"
s:Sc/IonsWithCharge3to5/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/IonsWithCharge3to5/OnlyIncludeParticlesNamed = 3 "GenericIon(6,12,3)" "GenericIon(6,12,4)" "GenericIon(6,12,5)"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "GenericIon(6,12)"
Origin.txt

# Particles pass through Film1Container which contains a divided box, Film1.
# We score fluence in a phantom, filtering several different ways based on
# origin of the scored particle.

s:Ge/Film1Container/Type = "TsBox"
s:Ge/Film1Container/Parent = "World"
s:Ge/Film1Container/Material = "G4_WATER"
d:Ge/Film1Container/HLX = 2.1 m
d:Ge/Film1Container/HLY = 2.1 m
d:Ge/Film1Container/HLZ = .15 m
d:Ge/Film1Container/TransZ = 1. m

s:Ge/Film1/Type = "TsBox"
s:Ge/Film1/Parent = "Film1Container"
s:Ge/Film1/Material = "G4_WATER"
d:Ge/Film1/HLX = 2. m
d:Ge/Film1/HLY = 2. m
d:Ge/Film1/HLZ = .1 m
i:Ge/Film1/XBins = 2
i:Ge/Film1/YBins = 2
i:Ge/Film1/ZBins = 2

s:Ge/Phantom/Type = "TsBox"
s:Ge/Phantom/Parent = "World"
s:Ge/Phantom/Material = "G4_WATER"
d:Ge/Phantom/HLX = 3. m
d:Ge/Phantom/HLY = 3. m
d:Ge/Phantom/HLZ = 0.5 m
d:Ge/Phantom/TransZ = -1. m
i:Ge/Phantom/XBins = 1
i:Ge/Phantom/YBins = 1
i:Ge/Phantom/ZBins = 1

s:Sc/ParticlesFromFilm1ContainerAndSubcomponents/Quantity = "Fluence"
s:Sc/ParticlesFromFilm1ContainerAndSubcomponents/Component = "Phantom"
s:Sc/ParticlesFromFilm1ContainerAndSubcomponents/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/ParticlesFromFilm1ContainerAndSubcomponents/OutputToConsole = "True"
s:Sc/ParticlesFromFilm1ContainerAndSubcomponents/OnlyIncludeParticlesFromComponentOrSubComponentsOf = 1 "Film1Container"
s:Sc/ParticlesFromFilm1ContainerButNotSubcomponents/Quantity = "Fluence"
s:Sc/ParticlesFromFilm1ContainerButNotSubcomponents/Component = "Phantom"
ParallelBoxRebinned.txt

# Score in a variety of x, y and z divisions.
# Topas creates parallel worlds as needed to accomplish this.

d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/MassWorldBox/Parent = "World"
s:Ge/MassWorldBox/Type = "Group"
s:Ge/MassWorldBox/Material = "G4_WATER"
b:Ge/MassWorldBox/IsParallel = "FALSE"
d:Ge/MassWorldBox/HLX = 9.0 cm
d:Ge/MassWorldBox/HLY = 9.0 cm
TOPAS Documentation, Release 3.0

```plaintext
d:Ge/MassWorldBox/HLZ = 9.0 cm
s:Ge/MassWorldBox/Color = "blue"

s:Ge/ParallelWorldBox/Parent = "MassWorldBox"
s:Ge/ParallelWorldBox/Type = "TsBox"
b:Ge/ParallelWorldBox/IsParallel = "TRUE"
d:Ge/ParallelWorldBox/HLX = 6.0 cm
d:Ge/ParallelWorldBox/HLY = 5.0 cm
d:Ge/ParallelWorldBox/HLZ = 4.0 cm
g:Ge/ParallelWorldBox/Color = "green"

i:Ge/ParallelWorldBox/XBins = 1
i:Ge/ParallelWorldBox/YBins = 2
i:Ge/ParallelWorldBox/ZBins = 3

i:Sc/Score2by3by4/Quantity = "DoseToMedium"
v:Sc/Score2by3by4/OnlyIncludeParticlesFromProcess = 1 "eBrem"

i:Sc/Score2by3by4/Component = "ParallelWorldBox"
b:Sc/Score2by3by4/OutputToConsole = "TRUE"
i:Sc/Score2by3by4/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/Score2by3by4/XBins = 2
i:Sc/Score2by3by4/YBins = 3
i:Sc/Score2by3by4/ZBins = 4

i:Sc/Score3by4by5/Quantity = "DoseToMedium"
v:Sc/Score3by4by5/OnlyIncludeParticlesFromProcess = 1 "hIoni"

i:Sc/Score3by4by5/Component = "ParallelWorldBox"
b:Sc/Score3by4by5/OutputToConsole = "TRUE"
i:Sc/Score3by4by5/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/Score3by4by5/XBins = 3
i:Sc/Score3by4by5/YBins = 4
i:Sc/Score3by4by5/ZBins = 5

i:Sc/Score4by5by6/Quantity = "DoseToMedium"
v:Sc/Score4by5by6/OnlyIncludeParticlesFromProcess = 1 "hIoni"

i:Sc/Score4by5by6/Component = "ParallelWorldBox"
b:Sc/Score4by5by6/OutputToConsole = "TRUE"
i:Sc/Score4by5by6/IfOutputFileAlreadyExists = "Overwrite"
i:Sc/Score4by5by6/XBins = 4
i:Sc/Score4by5by6/YBins = 5
i:Sc/Score4by5by6/ZBins = 6

i:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
g:Gr/ViewA/Projection = "Perspective"
g:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 50.
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

i:So/Example/Type = "Beam"
i:So/Example/Component = "BeamPosition"
i:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504

i:So/Example/BeamPositionDistribution = "Gaussian"
i:So/Example/BeamPositionCutOffShape = "Ellipse"
```

Chapter 20. Scoring
SplitByTimeFeature.txt

```plaintext
# Demonstrates how scoring can be split by a time feature.
# In this example, though the time feature ImageName doesn't really get used to load any images,
# the scoring is still split by this time feature.
# This means that separate scorers are created for each of the three values of ImageName, and each scorer is active only when the ImageName value matches the that scorer's required value.

Ge/Phantom/Type        = "TsBox"
Ge/Phantom/Parent      = "World"
Ge/Phantom/Material    = "G4_WATER"
Ge/Phantom/HLX         = 80.0 cm
Ge/Phantom/HLY         = 80.0 cm
Ge/Phantom/HLZ         = 80.0 cm
Ge/Phantom/TransX      = 0. cm
Ge/Phantom/TransY      = 0. cm
Ge/Phantom/TransZ      = -30. cm
Ge/Phantom/RotX        = 0. deg
Ge/Phantom/RotY        = 0. deg
Ge/Phantom/RotZ        = 0. deg

Sc/DoseAtPhantom/Quantity = "DoseToMedium"
Sc/DoseAtPhantom/Component = "Phantom"
Sc/DoseAtPhantom/OutputToConsole = "TRUE"
Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
Sc/DoseAtPhantom/SplitByTimeFeature = "ImageName"

# This time feature will take one of three string values.
Tf/ImageName/Function   = "Step"
Tf/ImageName/Times      = 3 10 20 30 ms
Tf/ImageName/Values     = 3 "lung1" "lung2" "lung3"
Tf/TimelineEnd          = 40.0 ms
Tf/NumberOfSequentialTimes = 4
Tf/Verbosity            = 1
Ph/Default/Modules      = 1 "g4em-standard_opt0"

So/Example/Type        = "Beam"
So/Example/Component   = "BeamPosition"
```

```
# TOPAS Documentation, Release 3.0

d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
```

```
Ts/ShowHistoryCountAtInterval = 10
Ts/PauseBeforeQuit = "True"
```

```
SplitByTimeFeature.txt
```

```
# TOPAS Documentation, Release 3.0

Tf/ImageName/Function = "Step"
Tf/ImageName/Times = 3 10 20 30 ms
Tf/ImageName/Values = 3 "lung1" "lung2" "lung3"
Tf/TimelineEnd = 40.0 ms
Ts/ShowHistoryCountAtInterval = 10
Ts/PauseBeforeQuit = "True"

i:Ts/ShowHistoryCountAtInterval = 10
b:Ts/PauseBeforeQuit = "True"
```
TOPAS Documentation, Release 3.0

Surfaces.txt

```plaintext
# Test of scoring on various surfaces

d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/TestBox/Type = "TsBox"
s:Ge/TestBox/Parent = "World"
s:Ge/TestBox/Material = "G4_WATER"
d:Ge/TestBox/HLX = 20.0 cm
d:Ge/TestBox/HLY = 15.0 cm
d:Ge/TestBox/HLZ = 10.0 cm
s:Ge/TestBox/Color = "blue"
	s:Ge/TestSphere/Type = "TsSphere"
s:Ge/TestSphere/Parent = "World"
s:Ge/TestSphere/Material = "G4_WATER"
d:Ge/TestSphere/RMin = 5.0 cm
d:Ge/TestSphere/RMax = 7.0 cm
d:Ge/TestSphere/SPhi = 0. deg
d:Ge/TestSphere/DPhi = 360. deg
d:Ge/TestSphere/STheta = 0. deg
d:Ge/TestSphere/DTeta = 180. deg
d:Ge/TestSphere/TransZ = -20. cm
s:Ge/TestSphere/Color = "blue"
	s:Ge/TestCylinder/Type = "TsCylinder"
s:Ge/TestCylinder/Parent = "World"
s:Ge/TestCylinder/Material = "G4_WATER"
d:Ge/TestCylinder/RMin = 0.0 cm
d:Ge/TestCylinder/RMax = 8.0 cm
d:Ge/TestCylinder/HL = 10.0 cm
d:Ge/TestCylinder/SPhi = 0. deg
d:Ge/TestCylinder/DPhi = 360. deg
```
d:Ge/TestCylinder/TransZ = -40. cm
s:Ge/TestCylinder/Color = "blue"

s:Sc/BoxXPPlus/Quantity = "SurfaceTrackCount"
s:Sc/BoxXPPlus/Surface = "TestBox/XPlusSurface"
b:Sc/BoxXPPlus/OutputToConsole = "TRUE"

s:Sc/BoxXMinus/Quantity = "SurfaceTrackCount"
s:Sc/BoxXMinus/Surface = "TestBox/XMinusSurface"
b:Sc/BoxXMinus/OutputToConsole = "TRUE"

s:Sc/BoxYPPlus/Quantity = "SurfaceTrackCount"
s:Sc/BoxYPPlus/Surface = "TestBox/YPlusSurface"
b:Sc/BoxYPPlus/OutputToConsole = "TRUE"

s:Sc/BoxYMinus/Quantity = "SurfaceTrackCount"
s:Sc/BoxYMinus/Surface = "TestBox/YMinusSurface"
b:Sc/BoxYMinus/OutputToConsole = "TRUE"

s:Sc/BoxZPPlus/Quantity = "SurfaceTrackCount"
s:Sc/BoxZPPlus/Surface = "TestBox/ZPlusSurface"
b:Sc/BoxZPPlus/OutputToConsole = "TRUE"

s:Sc/BoxZMinus/Quantity = "SurfaceTrackCount"
s:Sc/BoxZMinus/Surface = "TestBox/ZMinusSurface"
b:Sc/BoxZMinus/OutputToConsole = "TRUE"

s:Sc/SphereInner/Quantity = "SurfaceTrackCount"
s:Sc/SphereInner/Surface = "TestSphere/InnerCurvedSurface"
b:Sc/SphereInner/OutputToConsole = "TRUE"

s:Sc/SphereOuter/Quantity = "SurfaceTrackCount"
s:Sc/SphereOuter/Surface = "TestSphere/OuterCurvedSurface"
b:Sc/SphereOuter/OutputToConsole = "TRUE"

s:Sc/CylinderZPPlus/Quantity = "SurfaceTrackCount"
s:Sc/CylinderZPPlus/Surface = "TestCylinder/ZPlusSurface"
b:Sc/CylinderZPPlus/OutputToConsole = "TRUE"

s:Sc/CylinderZMinus/Quantity = "SurfaceTrackCount"
s:Sc/CylinderZMinus/Surface = "TestCylinder/ZMinusSurface"
b:Sc/CylinderZMinus/OutputToConsole = "TRUE"

s:Sc/CylinderInner/Quantity = "SurfaceTrackCount"
s:Sc/CylinderInner/Surface = "TestCylinder/InnerCurvedSurface"
b:Sc/CylinderInner/OutputToConsole = "TRUE"

s:Sc/CylinderOuter/Quantity = "SurfaceTrackCount"
s:Sc/CylinderOuter/Surface = "TestCylinder/OuterCurvedSurface"
b:Sc/CylinderOuter/OutputToConsole = "TRUE"

s:Gr/ViewA/Type = "OpenGL"
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
s:So/Example/BeamEnergy = 169.23 MeV
TOPAS Documentation, Release 3.0

$\text{So/Example/BeamEnergySpread} = 0.757504$
$\text{So/Example/BeamPositionDistribution} = \text{"Gaussian"}$
$\text{So/Example/BeamPositionCutoffShape} = \text{"Ellipse"}$
$\text{So/Example/BeamPositionCutoffX} = 10. \text{ cm}$
$\text{So/Example/BeamPositionCutoffY} = 10. \text{ cm}$
$\text{So/Example/BeamPositionSpreadX} = 0.65 \text{ cm}$
$\text{So/Example/BeamPositionSpreadY} = 0.65 \text{ cm}$
$\text{So/Example/BeamAngularDistribution} = \text{"Gaussian"}$
$\text{So/Example/BeamAngularCutoffX} = 90. \text{ deg}$
$\text{So/Example/BeamAngularCutoffY} = 90. \text{ deg}$
$\text{So/Example/BeamAngularSpreadX} = 0.0032 \text{ rad}$
$\text{So/Example/BeamAngularSpreadY} = 0.0032 \text{ rad}$
$\text{So/Example/NumberOfHistoriesInRun} = 1000$
$\text{Ts/ShowHistoryCountAtInterval} = 100$
$\text{Ts/PauseBeforeQuit} = \text{"True"}$
CHAPTER 21

PhaseSpace

MultiRun_Write_ASCII.txt

# Writes phase space separately for several time intervals
b:Ge/World/Invisible = "TRUE"
s:Ge/VacFilm/Type = "TsBox"
s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "G4_WATER"
d:Ge/VacFilm/HLX = 50.0 cm
d:Ge/VacFilm/HLY = 50.0 cm
d:Ge/VacFilm/HLZ = 1.0 cm
d:Ge/VacFilm/TransX = 0. cm
d:Ge/VacFilm/TransY = 0. cm
d:Ge/VacFilm/TransZ = 0. cm
d:Ge/VacFilm/RotX = 0. deg
d:Ge/VacFilm/RotY = 0. deg
d:Ge/VacFilm/RotZ = 0. deg
s:Ge/VacFilm/Color = "skyblue"
s:Sc/PhaseSpaceAtVacFilm/Quantity = "PhaseSpace"
b:Sc/PhaseSpaceAtVacFilm/OutputToConsole = "True"
s:Sc/PhaseSpaceAtVacFilm/Surface = "VacFilm/ZMinusSurface"
s:Sc/PhaseSpaceAtVacFilm/OutputType = "ASCII" # ASCII or Binary
t:Sc/PhaseSpaceAtVacFilm/OutputFile = "ASCIIOutput"
i:Sc/PhaseSpaceAtVacFilm/OutputBufferSize = 1000
#s:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesGoing = "In"
b:Sc/PhaseSpaceAtVacFilm/IncludeRunID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeEventID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTrackID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTime = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeSeed = "True"
#sv:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesNamed = 1 "Proton"
ReadASCII.txt

# Read phase space in TOPAS ASCII form

b:Ge/World/Invisible = "TRUE"

s:Ge/VacFilm/Type = "TsBox"
s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "G4_WATER"
d:Ge/VacFilm/HLX = 50.0 cm
d:Ge/VacFilm/HLY = 50.0 cm
d:Ge/VacFilm/HLZ = 1.0 cm
d:Ge/VacFilm/TransX = 0. cm
d:Ge/VacFilm/TransY = 0. cm
d:Ge/VacFilm/TransZ = 0. cm
d:Ge/VacFilm/RotX = 0. deg
d:Ge/VacFilm/RotY = 0. deg
d:Ge/VacFilm/RotZ = 0. deg
ReadBinary.txt

# Read phase space in TOPAS Binary form

b:Ge/World/Invisible = "TRUE"

a:Ge/VacFilm/Type = "TsBox"
a:Ge/VacFilm/Parent = "World"
a:Ge/VacFilm/Material = "G4_WATER"
a:Ge/VacFilm/HLX = 50.0 cm
a:Ge/VacFilm/HLY = 50.0 cm
a:Ge/VacFilm/HLZ = 1.0 cm
a:Ge/VacFilm/TransX = 0. cm
a:Ge/VacFilm/TransY = 0. cm
a:Ge/VacFilm/TransZ = 0. cm
a:Ge/VacFilm/RotX = 0. deg
a:Ge/VacFilm/RotY = 0. deg
a:Ge/VacFilm/RotZ = 0. deg
a:Ge/VacFilm/Color = "skyblue"

a:So/Example/Type                  = "PhaseSpace"
a:So/Example/PhaseSpaceFileName    = "BinaryOutput"
a:So/Example/Component            = "World"
#i:So/Example/PhaseSpaceMultipleUse = 2
#i:So/Example/PhaseSpaceBufferSize = 100
b:So/Example/PhaseSpaceIncludeEmptyHistories = "True"

# Graphics
a:Gr/ViewA/Type               = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta       = 55 deg
d:Gr/ViewA/Phi         = 20 deg
d:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom    = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

i:Ts/ShowHistoryCountAtInterval = 10
b:Ts/PauseBeforeQuit = "True"
ReadLimited.txt

# Write phase space in Limited form

b: Ge/World/Invisible = "TRUE"

s: Ge/VacFilm/Type = "TsBox"
s: Ge/VacFilm/Parent = "World"
s: Ge/VacFilm/Material = "G4_WATER"
d: Ge/VacFilm/HLX = 50.0 cm
d: Ge/VacFilm/HLY = 50.0 cm
d: Ge/VacFilm/HLZ = 1.0 cm
d: Ge/VacFilm/TransX = 0. cm
d: Ge/VacFilm/TransY = 0. cm
d: Ge/VacFilm/TransZ = 0. cm
d: Ge/VacFilm/RotX = 0. deg
d: Ge/VacFilm/RotY = 0. deg
d: Ge/VacFilm/RotZ = 0. deg
s: Ge/VacFilm/Color = "skyblue"

s: So/Example/Type = "PhaseSpace"
s: So/Example/PhaseSpaceFileName = "LimitedOutput"
s: So/Example/Component = "World"
#i: So/Example/PhaseSpaceMultipleUse = 2
#i: So/Example/PhaseSpaceBufferSize = 100
b: So/Example/PhaseSpaceIncludeEmptyHistories = "True"

# Graphics
s: Gr/ViewA/Type = "OpenGL"
i: Gr/ViewA/WindowSizeX = 900
i: Gr/ViewA/WindowSizeY = 900
d: Gr/ViewA/Theta = 55 deg
d: Gr/ViewA/Phi = 20 deg
s: Gr/ViewA/Projection = "Perspective"
d: Gr/ViewA/PerspectiveAngle = 30 deg
u: Gr/ViewA/Zoom = 1.3
b: Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

i: Ts/ShowHistoryCountAtInterval = 10
b: Ts/PauseBeforeQuit = "True"
WriteASCII.txt

# Write phase space in TOPAS ASCII form

b:Ge/World/Invisible = "TRUE"

s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "G4_WATER"
d:Ge/VacFilm/HLX = 50.0 cm
d:Ge/VacFilm/HLY = 50.0 cm
d:Ge/VacFilm/HLZ = 1.0 cm
d:Ge/VacFilm/TransX = 0. cm
d:Ge/VacFilm/TransY = 0. cm
d:Ge/VacFilm/TransZ = 0. cm
d:Ge/VacFilm/RotX = 0. deg
d:Ge/VacFilm/RotY = 0. deg
d:Ge/VacFilm/RotZ = 0. deg
s:Ge/VacFilm/Color = "skyblue"

s:Sc/PhaseSpaceAtVacFilm/Quantity = "PhaseSpace"
b:Sc/PhaseSpaceAtVacFilm/OutputToConsole = "True"
s:Sc/PhaseSpaceAtVacFilm/Surface = "VacFilm/ZMinusSurface"
s:Sc/PhaseSpaceAtVacFilm/OutputType = "ASCII"  # ASCII, Binary, Limited or ROOT

b:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesGoing = "In"
b:Sc/PhaseSpaceAtVacFilm/IncludeTOPASTime = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTimeOfFlight = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeRunID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeEventID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTrackID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeParentID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeVertexInfo = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeSeed = "True"
s:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesNamed = 1 "Proton"

s:Sc/PhaseSpaceAtVacFilm/IfOutputFileAlreadyExists = "Overwrite"

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
w:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
WriteBinary.txt

Write phase space in TOPAS Binary form

```plaintext
# Write phase space in TOPAS Binary form

b:Ge/World/Invisible = "TRUE"

s:Ge/VacFilm/Type = "TsBox"
s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "G4_WATER"
d:Ge/VacFilm/HLX = 50.0 cm
d:Ge/VacFilm/HLY = 50.0 cm
d:Ge/VacFilm/HLZ = 1.0 cm
d:Ge/VacFilm/TransX = 0. cm
d:Ge/VacFilm/TransY = 0. cm
d:Ge/VacFilm/TransZ = 0. cm
d:Ge/VacFilm/RotX = 0. deg
d:Ge/VacFilm/RotY = 0. deg
d:Ge/VacFilm/RotZ = 0. deg
s:Ge/VacFilm/Color = "skyblue"

s:Sc/PhaseSpaceAtVacFilm/Quantity = "PhaseSpace"
bs:Sc/PhaseSpaceAtVacFilm/OutputToConsole = "True"
s:Sc/PhaseSpaceAtVacFilm/Surface = "VacFilm/ZMinusSurface"
s:Sc/PhaseSpaceAtVacFilm/OutputType = "Binary" # ASCII, Binary, ...
    Limited or ROOT
s:Sc/PhaseSpaceAtVacFilm/OutputFile = "BinaryOutput"
i:Sc/PhaseSpaceAtVacFilm/OutputBufferSize = 1000
#s:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesGoing = "In"
b:Sc/PhaseSpaceAtVacFilm/IncludeTOPASTime = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTimeOfFlight = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeRunID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeEventID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTrackID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeParentID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeVertexInfo = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeSeed = "True"
#s:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesNamed = 1 "Proton"
s:Sc/PhaseSpaceAtVacFilm/IfOutputFileAlreadyExists = "Overwrite"

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta = 55 deg
```

Chapter 21. PhaseSpace
d:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffXX = 10. cm
d:So/Example/BeamPositionCutoffYY = 10. cm
d:So/Example/BeamPositionSpreadXX = 0.65 cm
d:So/Example/BeamPositionSpreadYY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffXX = 90. deg
d:So/Example/BeamAngularCutoffYY = 90. deg
d:So/Example/BeamAngularSpreadXX = 0.0032 rad
d:So/Example/BeamAngularSpreadYY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 100

i:Ts/ShowHistoryCountAtInterval = 10
b:Ts/PauseBeforeQuit = "True"

WriteIonsASCII.txt

# Write ASCII phase space with ion beam

b:Ge/World/Invisible = "TRUE"

s:Ge/VacFilm/Type = "TsBox"
s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "Vacuum"
d:Ge/VacFilm/HLX = 50.0 cm
d:Ge/VacFilm/HLY = 50.0 cm
d:Ge/VacFilm/HLZ = 1.0 cm
d:Ge/VacFilm/TransX = 0. cm
d:Ge/VacFilm/TransY = 0. cm
d:Ge/VacFilm/TransZ = 0. cm
d:Ge/VacFilm/RotX = 0. deg
d:Ge/VacFilm/RotY = 0. deg
d:Ge/VacFilm/RotZ = 0. deg
s:Ge/VacFilm/Color = "skyblue"

s:Sc/PhaseSpaceAtVacFilm/Quantity = "PhaseSpace"
b:Sc/PhaseSpaceAtVacFilm/OutputToConsole = "True"
s:Sc/PhaseSpaceAtVacFilm/Surface = "VacFilm/ZMinusSurface"
s:Sc/PhaseSpaceAtVacFilm/OutputType = "ASCII" # ASCII, Binary, Limited or ROOT
s:Sc/PhaseSpaceAtVacFilm/OutputFile = "ASCIIOutput"
i:Sc/PhaseSpaceAtVacFilm/OutputBufferSize = 1000
sv:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesNamed = 1 "GenericIon(3,7)"

21.7. WriteIonsASCII.txt
WriteLimited.txt

```plaintext
# Write phase space in Limited form

b:Ge/World/Invisible = "TRUE"

s:Ge/VacFilm/Type = "TsBox"

s:Ge/VacFilm/Parent = "World"

s:Ge/VacFilm/Material = "G4_WATER"

d:Ge/VacFilm/HLX = 50.0 cm

d:Ge/VacFilm/HLY = 50.0 cm

d:Ge/VacFilm/HLZ = 1.0 cm

d:Ge/VacFilm/TransX = 0. cm

d:Ge/VacFilm/TransY = 0. cm

d:Ge/VacFilm/TransZ = 0. cm

d:Ge/VacFilm/RotX = 0. deg

d:Ge/VacFilm/RotY = 0. deg

d:Ge/VacFilm/RotZ = 0. deg

s:Ge/VacFilm/Color = "skyblue"
```
WriteROOT.txt

# Write phase space in TOPAS ASCII form

b:Ge/World/Invisible = "TRUE"

WriteROOT.txt
s:Ge/VacFilm/Type       = "TsBox"
s:Ge/VacFilm/Parent     = "World"
s:Ge/VacFilm/Material   = "G4_WATER"
d:Ge/VacFilm/HLX       = 50.0 cm
d:Ge/VacFilm/HLY       = 50.0 cm
d:Ge/VacFilm/HLZ       = 1.0 cm
d:Ge/VacFilm/TransX    = 0. cm
d:Ge/VacFilm/TransY    = 0. cm
d:Ge/VacFilm/TransZ    = 0. cm
d:Ge/VacFilm/RotX      = 0. deg
d:Ge/VacFilm/RotY      = 0. deg
d:Ge/VacFilm/RotZ      = 0. deg
s:Ge/VacFilm/Color     = "skyblue"
s:Sc/PhaseSpaceAtVacFilm/Quantity = "PhaseSpace"
b:Sc/PhaseSpaceAtVacFilm/OutputToConsole = "True"
s:Sc/PhaseSpaceAtVacFilm/Surface   = "VacFilm/ZMinusSurface"
s:Sc/PhaseSpaceAtVacFilm/OutputType = "ROOT" # ASCII, Binary, ...
s:Sc/PhaseSpaceAtVacFilm/OutputBufferSize = 1000
#s:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesGoing = "In"
b:Sc/PhaseSpaceAtVacFilm/IncludeTOPASTime = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTimeOfFlight = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeEventID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeTrackID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeParentID = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeVertexInfo = "True"
b:Sc/PhaseSpaceAtVacFilm/IncludeSeed = "True"
sv:Sc/PhaseSpaceAtVacFilm/OnlyIncludeParticlesNamed = 1 "Proton"
s:Sc/PhaseSpaceAtVacFilm/IfOutputFileAlreadyExists = "Overwrite"
s:Gr/ViewA/Type        = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
i:Gr/ViewA/WindowSizeY = 900
d:Gr/ViewA/Theta       = 55 deg
d:Gr/ViewA/Phi         = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom        = 1.3
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

s:So/Example/Type      = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
d:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX  = 0.0032 rad
d:So/Example/BeamAngularSpreadY  = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun  = 100

i:Ts/ShowHistoryCountAtInterval  = 10
b:Ts/PauseBeforeQuit  = "True"
A box undergoes rotation relative to a parent box which is itself undergoing a different rotation.

d:Ge/World/HLX = 10 m
d:Ge/World/HLY = 10 m
d:Ge/World/HLZ = 10 m
s:Ge/World/Material = "Vacuum"
b:Ge/World/Invisible = "True"

d:Ge/BeamPosition/TransX = 3.9 m
d:Ge/BeamPosition/TransY = 3.9 m

s:Ge/OuterBox/Parent = "World"
s:Ge/OuterBox/Type = "TsBox"
s:Ge/OuterBox/Material = "Air"
d:Ge/OuterBox/HLX = 4 m
d:Ge/OuterBox/HLY = 4 m
d:Ge/OuterBox/HLZ = 4 m
d:Ge/OuterBox/TransX = 2 m
d:Ge/OuterBox/TransY = 2 m
d:Ge/OuterBox/TransZ = 0 m
d:Ge/OuterBox/RotX = 0 deg
d:Ge/OuterBox/RotY = 0 deg
d:Ge/OuterBox/RotZ = Tf/OuterBoxStep/Value deg

s:Ge/InnerBox/Parent = "OuterBox"
s:Ge/InnerBox/Type = "TsBox"
s:Ge/InnerBox/Material = "Lead"
d:Ge/InnerBox/HLX = 1 m
d:Ge/InnerBox/HLY = 1 m
d:Ge/InnerBox/HLZ = 2 m
\begin{verbatim}
Ge/InnerBox/TransX = 1. m
Ge/InnerBox/TransY = 1. m
Ge/InnerBox/TransZ = 0. m
Ge/InnerBox/RotX = 0. deg
Ge/InnerBox/RotY = Tf/InnerBoxStep/Value deg
Ge/InnerBox/RotZ = Tf/InnerBoxStep/Value deg
Ge/InnerBox/DrawingStyle = "Solid"
Ge/InnerBox/Color = "red"

Tf/OuterBoxRot/Function = "Linear deg"
Tf/OuterBoxRot/Rate = 2. deg/ms
Tf/OuterBoxRot/StartValue = -28.0 deg
Tf/OuterBoxRot/RepetitionInterval = 360. ms

Tf/InnerBoxRot/Function = "Linear deg"
Tf/InnerBoxRot/Rate = -2. deg/ms
Tf/InnerBoxRot/StartValue = 0.0 deg
Tf/InnerBoxRot/RepetitionInterval = 360. ms

Tf/InnerBoxStep/Function = "Step"
Tv/Tf/InnerBoxStep/Times = 2 6 60 ms
Tv/Tf/InnerBoxStep/Values = 2.0 Tf/InnerBoxRot/Value deg
Tf/InnerBoxStep/RepetitionInterval = 360. ms

Tf/OuterBoxStep/Function = "Step"
Tv/Tf/OuterBoxStep/Times = 2 14 60 ms
Tv/Tf/OuterBoxStep/Values = 2.0 Tf/OuterBoxRot/Value deg
Tf/OuterBoxStep/RepetitionInterval = 360. ms

Tv/Verbosity = 1
Tv/TimelineEnd = 30.0 ms
Tv/NumberOfSequentialTimes = 40

Gr/ViewA/Type = "OpenGL"
Gr/ViewA/WindowSizeX = 1024
Gr/ViewA/WindowSizeY = 768
Gr/ViewA/Theta = 55 deg
Gr/ViewA/Phi = 20 deg
Gr/ViewA/Zoom = 1.3
Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
Gr/ViewA/Projection = "Perspective"
Gr/ViewA/PerspectiveAngle = 30 deg
Gr/ViewA/CopyOpenGLToEPS = "False"

Ph/Default/Modules = 1 "g4em-standard_opt0"

So/Example/Type = "Beam"
So/Example/Component = "BeamPosition"
So/Example/BeamParticle = "proton"
So/Example/BeamEnergy = 169.23 MeV
So/Example/BeamEnergySpread = 0.757504
So/Example/BeamPositionDistribution = "Gaussian"
So/Example/BeamPositionCutoffShape = "Ellipse"
So/Example/BeamPositionCutoffX = 10. cm
So/Example/BeamPositionCutoffY = 10. cm
So/Example/BeamPositionSpreadX = 0.65 cm
So/Example/BeamPositionSpreadY = 0.65 cm
So/Example/BeamAngularDistribution = "Gaussian"
\end{verbatim}
CameraRotateAndZoom.txt

* Graphics views zoom and rotate under control of time features

```plaintext
s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
d:Ge/Box/TransX = 0. m
d:Ge/Box/TransY = 0. m
d:Ge/Box/TransZ = 0. m
d:Ge/Box/RotX = 0. deg
d:Ge/Box/RotY = 0. deg
d:Ge/Box/RotZ = 0. deg
s:Ge/Box/DrawingStyle = "Solid"

i:Gr/ViewA/WindowSizeX = 400
i:Gr/ViewA/WindowSizeY = 400
i:Gr/ViewA/WindowPosX = 0
i:Gr/ViewA/WindowPosY = 0
s:Gr/ViewA/Type = "OpenGL"
s:Gr/ViewA/ColorBy = "particletype"
s:v:Gr/ViewA/ColorByParticleTypeNames = 2 "proton" "e-"
s:v:Gr/ViewA/ColorByParticleTypeColors = 2 "red" "green"
d:Gr/ViewA/Theta = Tf/Rotate/Value deg
d:Gr/ViewA/Phi = Tf/Rotate/Value deg

s:Gr/ViewB/Type = "OpenGL"
i:Gr/ViewB/WindowSizeX = 400
i:Gr/ViewB/WindowSizeY = 400
i:Gr/ViewB/WindowPosX = 0
i:Gr/ViewB/WindowPosY = 400
u:Gr/ViewB/Zoom = Tf/Zoom/Value
b:Gr/ViewB/HiddenLineRemovalForTrajectories = "t"
s:Gr/ViewB/ColorBy = "origincomponent"
s:v:Gr/ViewB/ColorByOriginComponentNames = 2 "World" "Box"
s:v:Gr/ViewB/ColorByOriginComponentColors = 2 "yellow" "red"

s:Tf/Rotate/Function = "Linear deg"
d:Tf/Rotate/Rate = 2. deg/ms
d:Tf/Rotate/StartValue = 0.0 deg
s:Tf/Rotate/RepetitionInterval = 360. ms

s:Tf/Zoom/Function = "Linear"
d:Tf/Zoom/Rate = 0.1 1/ms
u:Tf/Zoom/StartValue = 1
s:Tf/Zoom/RepetitionInterval = 360. ms
```
ChangingKEFilterByTimeFeature.txt

# Kinetic Energy filter cutoff varies by time feature

b:Ts/PauseBeforeQuit = "True"
b:Ge/CheckForUnusedComponents = "False"

s:Ge/Box/Type = "TsBox"
s:Ge/Box/Parent = "World"
s:Ge/Box/Material = "G4_WATER"
d:Ge/Box/HLX = 1. m
d:Ge/Box/HLY = 1. m
d:Ge/Box/HLZ = .4 m
d:Ge/Box/TransX = 0. m
d:Ge/Box/TransY = 0. m
d:Ge/Box/TransZ = 0. m
d:Ge/Box/RotX = 0. deg
d:Ge/Box/RotY = 0. deg
d:Ge/Box/RotZ = 0. deg

s:Sc/Box/Quantity = "Fluence"
s:Sc/Box/Component = "Box"
s:Sc/Box/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/Box/OutputToConsole = "True"

s:Sc/Boxb/Quantity = "Fluence"
s:Sc/Boxb/Component = "Box"
s:Sc/Boxb/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/Boxb/OutputToConsole = "True"
ColorChange.txt

# Demonstrates step time feature by changing a box's color over time

d:Ge/World/HLX = 10. m
d:Ge/World/HLY = 10. m
d:Ge/World/HLZ = 10. m
b:Ge/World/Invisible = "TRUE"
s:Ge/BigBox/Parent = "World"
s:Ge/BigBox/Type = "TsBox"
s:Ge/BigBox/Material = "Air"
d:Ge/BigBox/HLX = 2. m
d:Ge/BigBox/HLY = 3. m
d:Ge/BigBox/HLZ = 4. m
CylinderGrowingInPhi.txt

* Demonstrate time features by changing a cylinder's phi extent and color over time.

d:Ge/World/HLX = 10. m
d:Ge/World/HLY = 10. m

CylinderGrowingInPhi.txt

* Demonstrate time features by changing a cylinder's phi extent and color over time.

d:Ge/World/HLX = 10. m
d:Ge/World/HLY = 10. m
d:Ge/World/HLZ = 10. m
b:Ge/World/Invisible = "True"

s:Ge/Outer/Type = "TsCylinder"
s:Ge/Outer/Parent = "World"
s:Ge/Outer/Material = "Vacuum"
d:Ge/Outer/RMin = 0.0 cm
d:Ge/Outer/RMax = 60.0 cm
d:Ge/Outer/SPhi = 0.0 deg
d:Ge/Outer/DPhi = 360. deg
d:Ge/Outer/HL = 3. cm
d:Ge/Outer/TransX = 0. cm
d:Ge/Outer/TransY = 0. cm
d:Ge/Outer/TransZ = -53. cm
d:Ge/Outer/RotX = 0. deg
d:Ge/Outer/RotY = 0. deg
d:Ge/Outer/RotZ = 0. deg
s:Ge/Outer2/Type = "TsCylinder"
s:Ge/Outer2/Parent = "World"
s:Ge/Outer2/Material = "Vacuum"
d:Ge/Outer2/RMin = 0.0 cm
d:Ge/Outer2/RMax = 60.0 cm
d:Ge/Outer2/SPhi = 0.0 deg
d:Ge/Outer2/DPhi = 360. deg
d:Ge/Outer2/HL = 3. cm
d:Ge/Outer2/TransX = 0. cm
d:Ge/Outer2/TransY = 0. cm
d:Ge/Outer2/TransZ = 53. cm
d:Ge/Outer2/RotX = 0. deg
d:Ge/Outer2/RotY = 0. deg
d:Ge/Outer2/RotZ = 0. deg
s:Ge/Cylinder/Type = "TsCylinder"
s:Ge/Cylinder/Parent = "World"
s:Ge/Cylinder/Material = "Vacuum"
d:Ge/Cylinder/RMin = 50.0 cm
d:Ge/Cylinder/RMax = 55.0 cm
d:Ge/Cylinder/SPhi = 20.0 deg
d:Ge/Cylinder/DPhi = Tf/Phi/Value deg
d:Ge/Cylinder/HL = 50. cm
d:Ge/Cylinder/TransX = 0. cm
d:Ge/Cylinder/TransY = 0. cm
d:Ge/Cylinder/TransZ = 0. cm
d:Ge/Cylinder/RotX = 0. deg
d:Ge/Cylinder/RotY = 0. deg
d:Ge/Cylinder/RotZ = 0. deg
s:Ge/Cylinder/Color = Tf/Color/Value
i:Ge/Cylinder/VisSegsPerCircle = 360
s:Ge/Cylinder/DrawingStyle = "FullWireFrame"

s:Tf/Color/Function = "Step"
sv:Tf/Color/Values = 4 "white" "blue" "green" "red"
dv:Tf/Color/Times = 4 1. 2. 3. 4. ms
s:Tf/PosPhi/Function = "Linear deg"
d:Tf/PosPhi/Rate = 12. deg/ms
d:Tf/PosPhi/StartValue = 10.0 deg


```
d:Tf/PosPhi/RepetitionInterval = 30. ms
s:Tf/NegPhi/Function = "Linear deg"
s:Tf/NegPhi/Rate = -12. deg/ms
s:Tf/NegPhi/StartValue = 360.0 deg
s:Tf/NegPhi/RepetitionInterval = 30. ms
s:Tf/Phi/Function = "Step"
dv:Tf/Phi/Times = 2 30 45 ms
dv:Tf/Phi/Values = 2 Tf/PosPhi/Value Tf/NegPhi/value deg
d:Tf/Phi/RepetitionInterval = 45. ms
i:Tf/Verbosity = 1
d:Tf/TimelineEnd = 38.0 ms
d:Tf/NumberOfSequentialTimes = 114

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
u:Gr/ViewA/Zoom = 2.
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
b:Gr/ViewA/CopyOpenGLToEPS = "False"

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
s:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
s:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
s:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10
```

Darkening.txt

```
* Boxes are set up such that they darken as they receive radiation.
* They therefore represent pieces of radiosensitive film.
* One of these films starts outside of the beam
* and moves into the beam over the course of the session.

s:Ge/World/Material = "Vacuum"
```

244 Chapter 22. TimeFeature
d:Ge/World/HLX = 10. m

d:Ge/World/HLY = 10. m

d:Ge/World/HLZ = 10. m

b:Ts/PauseBeforeQuit = "False"

s:So/Example/Type = "Beam"

s:So/Example/Component = "BeamPosition"

s:So/Example/BeamParticle = "proton"

s:So/Example/BeamEnergy = 169.23 MeV

s:So/Example/BeamEnergySpread = 0.757504

s:So/Example/BeamPositionDistribution = "Gaussian"

s:So/Example/BeamPositionCutoffShape = "Ellipse"

s:So/Example/BeamPositionCutoffX = 10. cm

s:So/Example/BeamPositionCutoffY = 10. cm

s:So/Example/BeamPositionSpreadX = 0.1 cm

s:So/Example/BeamPositionSpreadY = 0.1 cm

s:So/Example/BeamAngularDistribution = "Gaussian"

d:So/Example/BeamAngularCutoffX = 90. deg

d:So/Example/BeamAngularCutoffY = 90. deg

d:So/Example/BeamAngularSpreadX = 0.3 rad

d:So/Example/BeamAngularSpreadY = 0.0032 rad

i:So/Example/NumberOfHistoriesInRun = 4

d:Tf/BoxMoveDown/Function = "Linear m"

d:Tf/BoxMoveDown/Rate = -5. cm/ms

d:Tf/BoxMoveDown/StartValue = 1. m

d:Tf/BoxMoveDown/RepetitionInterval = 250. ms

s:Tf/BoxMove/Function = "Step"

dv:Tf/BoxMove/Times = 2 20. 250 ms

dv:Tf/BoxMove/Values = 2 Tf/BoxMoveDown/Value 0. cm

d:Tf/TimelineEnd = 75.0 ms

i:Tf/NumberOfSequentialTimes = 200

# Graphics
s:Gr/ViewA/Type = "OpenGL"

i:Gr/ViewA/WindowSizeX = 900

i:Gr/ViewA/WindowSizeY = 900

s:Gr/ViewA/Theta = 55 deg

s:Gr/ViewA/Phi = 20 deg

s:Gr/ViewA/Projection = "Perspective"

s:Gr/ViewA/PerspectiveAngle = 30 deg

s:Gr/ViewA/Zoom = 15.

s:Gr/ViewA/TransX = .0

s:Gr/ViewA/TransY = .2

b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

b:Gr/ViewA/CopyOpenGLToEPS = "True"

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

s:Ge/Phantom/Type = "TsBox"

s:Ge/Phantom/Parent = "World"

s:Ge/Phantom/Material = "G4_WATER"

d:Ge/Phantom/HLX = 30.0 cm
d:Ge/Phantom/HLY = 30.0 cm
d:Ge/Phantom/HLZ = 9. cm
d:Ge/Phantom/TransX = 0. cm
d:Ge/Phantom/TransZ = 20. cm
d:Ge/Phantom/RotX = 0. deg
d:Ge/Phantom/RotY = 0. deg
d:Ge/Phantom/RotZ = 0. deg
s:Ge/Phantom/Color = "white"
s:Ge/Phantom/DrawingStyle = "solid"
s:Ge/Phantom1/Type = "TsBox"
s:Ge/Phantom1/Parent = "World"
s:Ge/Phantom1/Material = "G4_WATER"
d:Ge/Phantom1/HLX = 30.0 cm
d:Ge/Phantom1/HLY = 30.0 cm
d:Ge/Phantom1/HLZ = 10.0 cm
d:Ge/Phantom1/TransX = 0. cm
d:Ge/Phantom1/TransY = 0. cm
d:Ge/Phantom1/TransZ = -30. cm
d:Ge/Phantom1/RotX = 0. deg
d:Ge/Phantom1/RotY = 0. deg
d:Ge/Phantom1/RotZ = 0. deg
s:Ge/Phantom1/Color = "white"
s:Ge/Phantom1/DrawingStyle = "solid"
s:Ge/Phantom2/Type = "TsBox"
s:Ge/Phantom2/Parent = "World"
s:Ge/Phantom2/Material = "G4_WATER"
d:Ge/Phantom2/HLX = 30.0 cm
d:Ge/Phantom2/HLY = 30.0 cm
d:Ge/Phantom2/HLZ = 10.0 cm
d:Ge/Phantom2/TransX = -70. cm
d:Ge/Phantom2/TransY = 0. cm
d:Ge/Phantom2/TransZ = -30. cm
d:Ge/Phantom2/RotX = 0. deg
d:Ge/Phantom2/RotY = 0. deg
d:Ge/Phantom2/RotZ = 0. deg
s:Ge/Phantom2/Color = "white"
s:Ge/Phantom2/DrawingStyle = "solid"
s:Ge/Phantom3/Type = "TsBox"
s:Ge/Phantom3/Parent = "World"
s:Ge/Phantom3/Material = "G4_WATER"
d:Ge/Phantom3/HLX = 30.0 cm
d:Ge/Phantom3/HLY = 30.0 cm
d:Ge/Phantom3/HLZ = 10.0 cm
d:Ge/Phantom3/TransX = 70. cm
d:Ge/Phantom3/TransY = 0. cm
d:Ge/Phantom3/TransZ = -30. cm
d:Ge/Phantom3/RotX = 0. deg
d:Ge/Phantom3/RotY = 0. deg
d:Ge/Phantom3/RotZ = 0. deg
s:Ge/Phantom3/Color = "white"
s:Ge/Phantom3/DrawingStyle = "solid"
s:Ge/Phantom4/Type = "TsBox"
s:Ge/Phantom4/Parent = "World"
s:Ge/Phantom4/Material = "G4_WATER"
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>d:Ge/Phantom4/HLX</td>
<td>30.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/HLY</td>
<td>30.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/HLZ</td>
<td>10.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/TransX</td>
<td>-140. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/TransY</td>
<td>0. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/TransZ</td>
<td>-30. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom4/RotX</td>
<td>0. deg</td>
</tr>
<tr>
<td>d:Ge/Phantom4/RotY</td>
<td>0. deg</td>
</tr>
<tr>
<td>d:Ge/Phantom4/RotZ</td>
<td>0. deg</td>
</tr>
<tr>
<td>s:Ge/Phantom4/Color</td>
<td>&quot;white&quot;</td>
</tr>
<tr>
<td>s:Ge/Phantom4/DrawingStyle</td>
<td>&quot;solid&quot;</td>
</tr>
<tr>
<td>s:Ge/Phantom5/Type</td>
<td>&quot;TsBox&quot;</td>
</tr>
<tr>
<td>s:Ge/Phantom5/Parent</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>s:Ge/Phantom5/Material</td>
<td>&quot;G4_WATER&quot;</td>
</tr>
<tr>
<td>d:Ge/Phantom5/HLX</td>
<td>30.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/HLY</td>
<td>30.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/HLZ</td>
<td>10.0 cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/TransX</td>
<td>140. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/TransY</td>
<td>0. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/TransZ</td>
<td>-30. cm</td>
</tr>
<tr>
<td>d:Ge/Phantom5/RotX</td>
<td>0. deg</td>
</tr>
<tr>
<td>d:Ge/Phantom5/RotY</td>
<td>0. deg</td>
</tr>
<tr>
<td>d:Ge/Phantom5/RotZ</td>
<td>0. deg</td>
</tr>
<tr>
<td>s:Ge/Phantom5/Color</td>
<td>&quot;white&quot;</td>
</tr>
<tr>
<td>s:Ge/Phantom5/DrawingStyle</td>
<td>&quot;solid&quot;</td>
</tr>
<tr>
<td>s:Sc/DoseAtPhantom/Quantity</td>
<td>&quot;EnergyDeposit&quot;</td>
</tr>
<tr>
<td>s:Sc/DoseAtPhantom/Component</td>
<td>&quot;Phantom&quot;</td>
</tr>
<tr>
<td>s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists</td>
<td>&quot;Overwrite&quot;</td>
</tr>
<tr>
<td>s:Sc/DoseAtPhantom/ColorBy</td>
<td>&quot;Sum&quot;</td>
</tr>
<tr>
<td>sv:Sc/DoseAtPhantom/ColorNames</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>&quot;white&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey240&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey220&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey200&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey180&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey160&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey140&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey120&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey100&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey080&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey060&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey040&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;grey020&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;black&quot;</td>
</tr>
<tr>
<td>dv:Sc/DoseAtPhantom/ColorValues</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>0.</td>
</tr>
<tr>
<td></td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td>3000</td>
</tr>
<tr>
<td></td>
<td>4000</td>
</tr>
<tr>
<td></td>
<td>5000</td>
</tr>
<tr>
<td></td>
<td>6000</td>
</tr>
<tr>
<td></td>
<td>7000</td>
</tr>
<tr>
<td></td>
<td>8000</td>
</tr>
<tr>
<td></td>
<td>9000</td>
</tr>
</tbody>
</table>
!10000
!11000
!12000
MeV
b:Sc/DoseAtPhantom/OutputAfterRun = "True"

s:Sc/DoseAtPhantom1/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom1/Component = "Phantom1"
s:Sc/DoseAtPhantom1/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom1/ColorBy = "Sum"
sv:Sc/DoseAtPhantom1/ColorNames = 14
 "white"
 "grey240"
 "grey220"
 "grey200"
 "grey180"
 "grey160"
 "grey140"
 "grey120"
 "grey100"
 "grey080"
 "grey060"
 "grey040"
 "grey020"
 "black"

dv:Sc/DoseAtPhantom1/ColorValues = 13
 0.
 1000
 2000
 3000
 4000
 5000
 6000
 7000
 8000
 9000
10000
11000
12000
MeV
b:Sc/DoseAtPhantom1/OutputAfterRun = "True"

s:Sc/DoseAtPhantom2/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom2/Component = "Phantom2"
s:Sc/DoseAtPhantom2/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom2/ColorBy = "Sum"
sv:Sc/DoseAtPhantom2/ColorNames = 14
 "white"
 "grey240"
 "grey220"
 "grey200"
 "grey180"
 "grey160"
 "grey140"
 "grey120"
 "grey100"
 "grey080"
 "grey060"
 "grey040"
 "grey020"
"grey140"
"grey120"
"grey100"
"grey080"
"grey060"
"grey040"
"grey020"
"black"

dv:Sc/DoseAtPhantom2/ColorValues = 13
0.
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
MeV

b:Sc/DoseAtPhantom2/OutputAfterRun = "True"

g:Sc/DoseAtPhantom3/Quantity = "EnergyDeposit"
g:Sc/DoseAtPhantom3/Component = "Phantom3"
g:Sc/DoseAtPhantom3/IfOutputFileAlreadyExists = "Overwrite"
g:Sc/DoseAtPhantom3/ColorBy = "Sum"
gv:Sc/DoseAtPhantom3/ColorNames = 14
"white"
"grey240"
"grey220"
"grey200"
"grey180"
"grey160"
"grey140"
"grey120"
"grey100"
"grey080"
"grey060"
"grey040"
"grey020"
"black"

dv:Sc/DoseAtPhantom3/ColorValues = 13
0.
1000
2000
3000
4000
5000
6000
7000
8000

b:Sc/DoseAtPhantom3/OutputAfterRun = "True"

s:Sc/DoseAtPhantom4/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom4/Component = "Phantom4"
s:Sc/DoseAtPhantom4/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom4/ColorBy = "Sum"
sv:Sc/DoseAtPhantom4/ColorNames = 14
"white"
"grey240"
"grey220"
"grey200"
"grey180"
"grey160"
"grey140"
"grey120"
"grey100"
"grey080"
"grey060"
"grey040"
"grey020"
"black"

dv:Sc/DoseAtPhantom4/ColorValues = 13
0.
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
MeV

b:Sc/DoseAtPhantom4/OutputAfterRun = "True"

s:Sc/DoseAtPhantom5/Quantity = "EnergyDeposit"
s:Sc/DoseAtPhantom5/Component = "Phantom5"
s:Sc/DoseAtPhantom5/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom5/ColorBy = "Sum"
sv:Sc/DoseAtPhantom5/ColorNames = 14
"white"
"grey240"
"grey220"
"grey200"
"grey180"
"grey160"
"grey140"
"grey120"
"grey100"
"grey080"
"grey060"
"grey040"
"grey020"
"black"

dv:Sc/DoseAtPhantom5/ColorValues = 13
0.1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
MeV

b:Sc/DoseAtPhantom5/OutputAfterRun = "True"

Logo.txt

# Constructs the name TOPAS out of a variety of geometry components
# and sends particles through this geometry.

b:Ge/World/Invisible = "True"

-------- Letter T --------

s:Ge/Letter_T/Parent = "World"
s:Ge/Letter_T/Type = "Group"
d:Ge/Letter_T/TransX = 0. m
d:Ge/Letter_T/TransY = 0.5 m
d:Ge/Letter_T/TransZ = 0. m
d:Ge/Letter_T/RotX = 0. deg
d:Ge/Letter_T/RotY = 0. deg
d:Ge/Letter_T/RotZ = 0. deg
#b:Ge/Letter_T/Include = "False"

s:Ge/T_1st/Type = "TsBox"
s:Ge/T_1st/Material = "Vacuum"
s:Ge/T_1st/Parent = "Letter_T"
d:Ge/T_1st/HLX = 50.0 cm
d:Ge/T_1st/HLY = 10.0 cm
d:Ge/T_1st/HLZ = 10.0 cm
d:Ge/T_1st/TransX = 0. m
### Letter O

- **Parent**: World
- **Type**: Group
- **TransX**: 1.0 m
- **TransY**: 0. m
- **TransZ**: 0. m
- **RotX**: 0. deg
- **RotY**: 0. deg
- **RotZ**: 0. deg

# O_Left

- **Type**: G4Cons
- **Material**: Aluminum
- **Parent**: Letter_O

RM[in,max]1 : -Z surface
RM[in,max]2 : +Z surface
- **RMin1**: 38.0 cm
- **RMax1**: 45.0 cm
- **RMin2**: 45.0 cm
- **RMax2**: 50.0 cm
- **HL**: 5.0 cm

*Sphi*: angle from x axis
- **SPhi**: 90.0 deg
- **DPhi**: 180.0 deg

# O_Right

- **Type**: TsSphere
- **Material**: Carbon
- **Parent**: Letter_O

RM[in,max]1 : -Z surface
RM[in,max]2 : +Z surface

#-------- Letter O --------
### Ge/O_Right

- **RMin**: 45.0 cm
- **RMax**: 50.0 cm

#### Ge/O_Right

- **SPhi**: angle from x axis
- **SPhi**: 271.0 deg
- **DPhi**: 130.0 deg

- **STheta**: angle from Y rotation
- **STheta**: 90.0 deg
- **DTheta**: 45.0 deg

- **TransX**: 0. m
- **TransY**: 0. m
- **TransZ**: 0. m
- **RotX**: 0. deg
- **RotY**: 0. deg
- **RotZ**: 0. deg

### Letter P

- **Parent**: "World"
- **Type**: "Group"

#### Ge/Letter_P

- **TransX**: 1.7 m
- **TransY**: 0. m
- **TransZ**: 0. m
- **RotX**: 0. deg
- **RotY**: 0. deg
- **RotZ**: 0. deg

#### Ge/P_1st

- **Type**: "G4EllipticalTube"
- **Material**: "Lexan"

- **HLX**: 9.0 cm
- **HLY**: 7.0 cm
- **HLZ**: 25.0 cm
- **TransX**: 0.0 cm
- **TransY**: Ge/P_1st/HLZ cm
- **TransZ**: 0.0 cm
- **RotX**: 90.0 deg
- **RotY**: 0.0 deg
- **RotZ**: 0.0 deg

#### Ge/P_2nd

- **Type**: "G4Trd"
- **Material**: "Kapton"

- **HLX1**: 10.0 cm
- **HLX2**: 9.5 cm
- **HLZ**: 25.0 cm
- **TransX**: 0.0 cm
- **TransY**: -1.0 * Ge/P_2nd/HLZ cm
- **TransZ**: 0.0 cm
- **RotX**: 90.0 deg
- **RotY**: 0.0 deg
- **RotZ**: 0.0 deg

#### Ge/P_3rd

- **Type**: "G4Torus"
- **Material**: "Lucite"
s:Ge/P_3rd/Parent = "Letter_P"
d:Ge/P_3rd/RMin = 5.0 cm
d:Ge/P_3rd/RMax = 6.0 cm
d:Ge/P_3rd/RTor = 25.0 cm
d:Ge/P_3rd/SPhi = 255.0 deg
d:Ge/P_3rd/DPhi = 190.0 deg
d:Ge/P_3rd/TransX = -8.0 cm + Ge/P_3rd/RTor
d:Ge/P_3rd/TransY = Ge/P_3rd/RTor cm
d:Ge/P_3rd/TransZ = 0.0 cm
d:Ge/P_3rd/RotX = 0.0 deg
d:Ge/P_3rd/RotY = 0.0 deg
d:Ge/P_3rd/RotZ = 0.0 deg

#--------- Letter A --------
s:Ge/Letter_A/Parent = "World"
s:Ge/Letter_A/Type = "Group"
d:Ge/Letter_A/TransX = 2.7 m
d:Ge/Letter_A/TransY = 0. m
d:Ge/Letter_A/TransZ = 0. m
d:Ge/Letter_A/RotX = 0. deg
d:Ge/Letter_A/RotY = 0. deg
d:Ge/Letter_A/RotZ = 0. deg
b:Ge/Letter_A/Include = "False"

#HLZ, Theta, Phi, HLY1,
#HLX1, HLX2, HLX3, HLX4,
#Alp2, Alpl, HLY2,
s:Ge/A_1st/Type = "G4Para"
s:Ge/A_1st/Material = "Brass"
s:Ge/A_1st/Parent = "Letter_A"
d:Ge/A_1st/HLX = 9.0 cm
d:Ge/A_1st/HLY = 50.0 cm
d:Ge/A_1st/HLZ = 7.0 cm
d:Ge/A_1st/Alpha = 15.0 deg
d:Ge/A_1st/Theta = 0.0 deg
d:Ge/A_1st/Phi = 0.0 deg
d:Ge/A_1st/TransX = -2.7 * Ge/A_1st/HLX cm
d:Ge/A_1st/TransY = 0.0 cm
d:Ge/A_1st/TransZ = 0.0 cm
d:Ge/A_1st/RotX = 0.0 deg
d:Ge/A_1st/RotY = 0.0 deg
d:Ge/A_1st/RotZ = 0.0 deg

#Will Try to build with G4GTrap or G4RTrap
s:Ge/A_2nd/Type = "G4Para"
s:Ge/A_2nd/Material = "Mylar"
s:Ge/A_2nd/Parent = "Letter_A"
d:Ge/A_2nd/HLX = 9.0 cm
d:Ge/A_2nd/HLY = 50.0 cm
d:Ge/A_2nd/HLZ = 7.0 cm
d:Ge/A_2nd/Alpha = -15.0 deg
d:Ge/A_2nd/Theta = 0.0 deg
d:Ge/A_2nd/Phi = 0.0 deg
d:Ge/A_2nd/TransX = -1.0 * Ge/A_1st/TransX cm
d:Ge/A_2nd/TransY = 0.0 cm
d:Ge/A_2nd/TransZ = 0.0 cm
d:Ge/A_2nd/RotX = 0.0 deg
d:Ge/A_2nd/RotY = 0.0 deg
#------- Letter S -------
s:Ge/Letter_S/Parent = "World"
s:Ge/Letter_S/Type = "Group"
#d:Ge/Letter_S/TransX = 0.0 m
d:Ge/Letter_S/TransX = 3.7 m
d:Ge/Letter_S/TransY = 0. m
d:Ge/Letter_S/TransZ = 0. m
d:Ge/Letter_S/RotX = 0. deg
d:Ge/Letter_S/RotY = 0. deg
d:Ge/Letter_S/RotZ = -20. deg
#b:Ge/Letter_S/Include = "False"
s:Ge/S_1st/Type = "G4HPolycone"
s:Ge/S_1st/Material = "Copper"
s:Ge/S_1st/Parent = "Letter_S"
d:Ge/S_1st/PhiStart = 350.0 deg
d:Ge/S_1st/PhiTotal = 225.0 deg
dv:Ge/S_1st/Z = 6 8.0 4.0 4.0 -4.0 -4.0 -8.0 cm
dv:Ge/S_1st/RInner = 6 10.0 10.0 10.0 10.0 10.0 10.0 cm
dv:Ge/S_1st/ROuter = 6 18.0 18.0 30.0 30.0 18.0 18.0 cm
d:Ge/S_1st/TransX = 10.0 cm
d:Ge/S_1st/TransY = 22.0 cm
d:Ge/S_1st/TransZ = 0.0 cm
d:Ge/S_1st/RotX = 0.0 deg
d:Ge/S_1st/RotY = 0.0 deg
d:Ge/S_1st/RotZ = 0.0 deg
s:Ge/S_2nd/Type = "G4Hype"
s:Ge/S_2nd/Material = "Brass"
s:Ge/S_2nd/Parent = "Letter_S"
d:Ge/S_2nd/IR = 7.0 cm
d:Ge/S_2nd/OR = 10.0 cm
d:Ge/S_2nd/IS = 10.0 deg
d:Ge/S_2nd/OS = 20.0 deg
d:Ge/S_2nd/HLZ = 10.0 cm
d:Ge/S_2nd/TransX = 0.0 cm
d:Ge/S_2nd/TransY = 0.0 cm
d:Ge/S_2nd/TransZ = 0.0 cm
d:Ge/S_2nd/RotX = 90.0 deg
d:Ge/S_2nd/RotY = 35.0 deg
d:Ge/S_2nd/RotZ = 0.0 deg
#b:Ge/S_2nd/Include = "false"
s:Ge/S_3rd/Type = "G4HPolyhedra"
s:Ge/S_3rd/Material = "Kapton"
s:Ge/S_3rd/Parent = "Letter_S"
d:Ge/S_3rd/PhiStart = 170.0 deg
d:Ge/S_3rd/PhiTotal = 225.0 deg
i:Ge/S_3rd/NSides = 8
dv:Ge/S_3rd/Z = 6
   8.0  4.0  4.0 -4.0 -4.0 -4.0  -8.0 cm
dv:Ge/S_3rd/RInner = 6
   10.0  10.0  10.0  10.0  10.0  10.0  10.0 cm
dv:Ge/S_3rd/ROuter = 6
   18.0  18.0  30.0  30.0  18.0  18.0  18.0 cm
d:Ge/S_3rd/TransX = -10.0 cm
d:Ge/S_3rd/TransY = -23.0 cm
d:Ge/S_3rd/TransZ = 0.0 cm
d:Ge/S_3rd/RotX = 0.0 deg
d:Ge/S_3rd/RotY = 0.0 deg
d:Ge/S_3rd/RotZ = 0.0 deg

# Default Beam position (S)
s:Ge/BeamPosition/Parent = "World"
s:Ge/BeamPosition/Type = "Group"
d:Ge/BeamPosition/TransX = 0.15 m
d:Ge/BeamPosition/TransY = 0.0 m
d:Ge/BeamPosition/TransZ = 0.0 m
#flipped cause the beam flies 0 to +z.
d:Ge/BeamPosition/RotX = 0.0 deg
d:Ge/BeamPosition/RotY = 270.0 deg
d:Ge/BeamPosition/RotZ = 0.0 deg
b:Ge/CheckForOverlaps = "False"

s:Tf/BeamCurrent/Function = "Step"
dv:Tf/BeamCurrent/Times = 2 18 20 s
iv:Tf/BeamCurrent/Values = 2
   18  0
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
w:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10.0 cm
d:So/Example/BeamPositionCutoffY = 10.0 cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
s:So/Example/BeamAngularCutoffX = 90.0 deg
d:So/Example/BeamAngularCutoffY = 90.0 deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = Tf/BeamCurrent/Value
d:Tf/TimeLineEnd = 20 s
i:Tf/NumberOfSequentialTimes = 20
s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 800
i:Gr/ViewA/WindowSizeY = 800
s:Gr/ViewA/ColorBy = "particletype"
s:Gr/ViewA/AxesComponent = "Letter_O"
d:Gr/ViewA/AxesSize = 20.0 cm
RotateResizeRecolor.txt

% A box is set to rotate, grow and change color
% as a test of the time feature system.

d;Ge/World/HLX = 10. m
d;Ge/World/HLY = 10. m
d;Ge/World/HLZ = 10. m

s;Ge/Box/Parent = "World"
s;Ge/Box/Type = "TsBox"
s;Ge/Box/Material = "Air"
d;Ge/Box/HLX = Tf/Size/Value m
d;Ge/Box/HLY = Tf/Size/Value m
d;Ge/Box/HLZ = 4. m
d;Ge/Box/TransX = 0. m
d;Ge/Box/TransY = 0. m
d;Ge/Box/TransZ = 0. m
d;Ge/Box/RotX = 0. deg
d;Ge/Box/RotY = 0. deg
d;Ge/Box/RotZ = Tf/Box/Value deg
s;Ge/Box/Color = Tf/Color/Value

s;Tf/Box/Function = "Linear deg"
d;Tf/Box/Rate = 1. deg/ms
d;Tf/Box/StartValue = 0.0 deg
d;Tf/Box/RepetitionInterval = 60. ms

s;Tf/Size/Function = "Linear cm"
d;Tf/Size/Rate = 0.1 m/ms
d;Tf/Size/StartValue = 1.0 m
d;Tf/Size/RepetitionInterval = 60. ms

s;Tf/Color/Function = "Step"
sv;Tf/Color/Values = 4 "white" "blue" "green" "red"
dv;Tf/Color/Times = 4 1. 2. 3. 4. ms

i;Tf/Verbosity = 1
d;Tf/TimelineEnd = 60.0 ms
d;Tf/NumberOfSequentialTimes = 60

s;Sc/MyScorer/Quantity = "EnergyDeposit"
s;Sc/MyScorer/Component = "Box"
b;Sc/MyScorer/OutputToConsole = "1"
s;Sc/MyScorer/IfOutputFileAlreadyExists = "Overwrite"
s;Gr/ViewA/Type = "OpenGL"
Rotation.txt

# Demonstrates direction of rotations in TOPAS placements
# by rotating a simple structure three ways.
# First, it rotates in the X direction.
# Then, after resetting, it rotates in the Y direction.
# Then, after resetting, it rotates in the Z direction.

d:Ge/World/HLX = 1.1 cm
d:Ge/World/HLY = 1.1 cm
d:Ge/World/HLZ = 1.1 cm
s:Ge/World/Color = "black"
s:Ge/system/Parent = "World"
s:Ge/system/Type = "Group"
d:Ge/system/RotX = Tf/rotationX/Value deg
d:Ge/system/RotY = Tf/rotationY/Value deg
d:Ge/system/RotZ = Tf/rotationZ/Value deg

s:Ge/tet/Parent = "system"
s:Ge/tet/Material = "Air"
s:Ge/tet/Type = "G4Tet"
dv:Ge/tet/Anchor = 3 0 0 1 cm
dv:Ge/tet/P2 = 3 1 0 0 cm
```
dv:Ge/tet/P3 = 3 0 1 0 cm
dv:Ge/tet/P4 = 3 0 0 0 cm
s:Ge/tet/Color = "yellow"
#d:Ge/tet/RotX = Tf/rotationX/Value deg
#d:Ge/tet/RotY = Tf/rotationY/Value deg
#d:Ge/tet/RotZ = Tf/rotationZ/Value deg
s:Ge/pointz/Parent = "system"
s:Ge/pointz/Material = "Air"
s:Ge/pointz/Type = "G4Orb"
d:Ge/pointz/R = 0.2 mm
s:Ge/pointz/Color = "blue"
s:Ge/pointz/DrawingStyle = "solid"
d:Ge/pointz/TransZ = 1 cm
s:Ge/pointy/Parent = "system"
s:Ge/pointy/Material = "Air"
s:Ge/pointy/Type = "G4Orb"
d:Ge/pointy/R = 0.2 mm
s:Ge/pointy/Color = "grass"
s:Ge/pointy/DrawingStyle = "solid"
d:Ge/pointy/TransY = 1 cm
s:Ge/pointx/Parent = "system"
s:Ge/pointx/Material = "Air"
s:Ge/pointx/Type = "G4Orb"
d:Ge/pointx/R = 0.2 mm
s:Ge/pointx/Color = "red"
s:Ge/pointx/DrawingStyle = "solid"
d:Ge/pointx/TransX = 1 cm
s:tf/rotationX/Function = "Step"
dv:tf/rotationX/Times = 288
dv:tf/rotationX/Values = 288
```

229. Rotation.txt
### Chapter 22. TimeFeature

#### Example 1: Step Function

**s:**
```
Tf/rotationY/Function = "Step"
```

**dv:**
```
Tf/rotationY/Times = 288
```

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-23</td>
<td>0</td>
</tr>
<tr>
<td>24-28</td>
<td>5</td>
</tr>
<tr>
<td>29-33</td>
<td>6</td>
</tr>
<tr>
<td>34-38</td>
<td>7</td>
</tr>
<tr>
<td>39-43</td>
<td>8</td>
</tr>
<tr>
<td>44-48</td>
<td>9</td>
</tr>
<tr>
<td>49-53</td>
<td>10</td>
</tr>
<tr>
<td>54-58</td>
<td>11</td>
</tr>
<tr>
<td>59-63</td>
<td>12</td>
</tr>
<tr>
<td>64-68</td>
<td>13</td>
</tr>
</tbody>
</table>

### Example 2: Rotation Function

**s:**
```
Tf/rotationZ/Function = "Step"
```

**dv:**
```
Tf/rotationZ/Times = 288
```

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Rotation (deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-23</td>
<td>0</td>
</tr>
<tr>
<td>24-28</td>
<td>34</td>
</tr>
<tr>
<td>29-33</td>
<td>35</td>
</tr>
<tr>
<td>34-38</td>
<td>36</td>
</tr>
<tr>
<td>39-43</td>
<td>37</td>
</tr>
<tr>
<td>44-48</td>
<td>38</td>
</tr>
<tr>
<td>49-53</td>
<td>39</td>
</tr>
<tr>
<td>54-58</td>
<td>40</td>
</tr>
<tr>
<td>59-63</td>
<td>41</td>
</tr>
<tr>
<td>64-68</td>
<td>42</td>
</tr>
<tr>
<td>69-73</td>
<td>43</td>
</tr>
</tbody>
</table>

---

**TOPAS Documentation, Release 3.0**
### Include File

```
includeFile = RunSequential_Mode.txt
```

--- Timeline setting

```
b:Tf/RandomizeTimeDistribution = "True"
```

--- Time Feature for Probability

```
s:Tf/Weights/Function = "Step"
```

```
v:Tf/Weights/Times = 5 1 2 3 4 5 s
```

```
v:Tf/Weights/Values = 5 1.0 0.8 0.6 0.4 0.2
```

--- Assign particles
# Same number of total particles in RunSequential_Mode.txt

i:So/Example/NumberOfHistoriesInRandomJob = 3000

u:So/Example/ProbabilityOfUsingAGivenRandomTime = Tf/Weights/Value

i:Ts/ShowHistoryCountAtInterval = 0

# Same number of total particles in RunSequential_Mode.txt

--- Dose for DoseGrid

s:Sc/dose/OutputFile = "Dose_Rand"

--- Demonstration of TOPAS time modes: Sequential Time Mode

--- Time Feature for Energy and Number of particles

s:Tf/Energies/Function = "Step"
dv:Tf/Energies/Times = 5 1 2 3 4 5 s
dv:Tf/Energies/Values = 5 110.0 100.0 90.0 80.0 70.0 MeV

s:Tf/Particles/Function = "Step"
dv:Tf/Particles/Times = 5 1 2 3 4 5 s
iv:Tf/Particles/Values = 5 1000 800 600 400 200

--- Timeline setting

--- Assign particles

--- Change number of particles per time stamps

RunSequential_Mode.txt

--- Demonstration of TOPAS time modes: Sequential Time Mode

--- Time Feature for Energy and Number of particles

s:Tf/Energies/Function = "Step"
dv:Tf/Energies/Times = 5 1 2 3 4 5 s
dv:Tf/Energies/Values = 5 110.0 100.0 90.0 80.0 70.0 MeV

s:Tf/Particles/Function = "Step"
dv:Tf/Particles/Times = 5 1 2 3 4 5 s
iv:Tf/Particles/Values = 5 1000 800 600 400 200

--- Timeline setting

--- Assign particles

--- Change number of particles per time stamps

# Demostration of TOPAS time modes: Sequential Time Mode

--- Time Feature for Energy and Number of particles

--- Timeline setting

--- Assign particles

--- Change number of particles per time stamps

For Sequential Time Mode

i:Tf/NumberOfSequentialTimes = 5

--- Assign particles

--- Change number of particles per time stamps
# Total particles are 1000 + 800 + 600 + 400 + 200 = 300
i: So/Example/NumberOfHistoriesInRun = Tf/Particles/Value

# Energy changes
d: So/Example/BeamEnergy = Tf/Energies/Value MeV

# From 10.0 cm
d: Ge/BeamPosition/TransZ = 10.0 cm

# Water phantom target
s: Ge/waterphantom/Type = "TsBox"
s: Ge/waterphantom/Parent = "World"
s: Ge/waterphantom/Material = "G4_WATER"
d: Ge/waterphantom/HLX = 5 cm
d: Ge/waterphantom/HLY = 5 cm
d: Ge/waterphantom/HLZ = 6 cm
d: Ge/waterphantom/TransX = 0. cm
d: Ge/waterphantom/TransY = 0. cm
d: Ge/waterphantom/TransZ = -1.0 * Ge/waterphantom/HLZ cm
d: Ge/waterphantom/RotX = 0. deg
d: Ge/waterphantom/RotY = 0. deg
d: Ge/waterphantom/RotZ = 0. deg
i: Ge/waterphantom/XBins = 1
i: Ge/waterphantom/YBins = 1
i: Ge/waterphantom/ZBins = 1
s: Ge/waterphantom/Color = "red"

# Dose grid for 1D PDD
s: Ge/dosegrid/Type = "TsCylinder"
s: Ge/dosegrid/Parent = "waterphantom"
d: Ge/dosegrid/RMin = 0.0 cm
d: Ge/dosegrid/RMax = 5 cm
d: Ge/dosegrid/SPhi = 0.0 deg
d: Ge/dosegrid/DPhi = 360.0 deg
d: Ge/dosegrid/HL = Ge/waterphantom/HLZ cm
d: Ge/dosegrid/TransX = 0. cm
d: Ge/dosegrid/TransY = 0. cm
d: Ge/dosegrid/TransZ = 0. cm
d: Ge/dosegrid/RotX = 0. deg
d: Ge/dosegrid/RotY = 0. deg
d: Ge/dosegrid/RotZ = 0. deg
i: Ge/dosegrid/RBins = 1
i: Ge/dosegrid/PhiBins = 1
i: Ge/dosegrid/ZBins = 120
b: Ge/dosegrid/IsParallel = "T"

# Dose for DoseGrid
s: Sc/dose/Quantity = "DoseToMedium"
s: Sc/dose/Component = "dosegrid"
s: Sc/dose/OutputFile = "Dose_Seq"
s: Sc/dose/OutputType = "binary"
b: Sc/dose/OutputToConsole = "F"
### Chapter 22. TimeFeature

- `b:Sc/dose/Visualize = "F"`
- `s:Sc/dose/IfOutputFileAlreadyExists = "Overwrite"

--- Summary

- `b:Ts/ShowCPExitingTime = "tRuE"` #For checking case sensitivity
- `i:Ts/ShowHistoryCountAtInterval = 10000`
Chapter 23

DICOM

DoseTo4DCT.txt

# Demonstrates use of a DICOM file.
# You must unzip synthetic_lung.tar.bz2 before you run this example.
# Reads in and displays a simple test DICOM that represents
# three different time phases of a simplified lung phantom.
# This phantom consists of a tumor in a cylinder of water in a tube of air.

includeFile = HUtoMaterialSchneider.txt

# Geometry
s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

s:Ge/Patient/Parent = "World"
s:Ge/Patient/Material = "G4_WATER"
d:Ge/Patient/TransX = 0. cm # -8. cm
d:Ge/Patient/TransY = 0. cm # -8. cm
d:Ge/Patient/TransZ = 0. m
d:Ge/Patient/RotX = 90. deg
d:Ge/Patient/RotY = 0. deg
d:Ge/Patient/RotZ = 0. deg
s:Ge/Patient/Type = "TsDicomPatient"
s:Ge/Patient/DicomDirectory = Tf/ImageName/Value
#b:Ge/Patient/PreLoadAllMaterials = "True"

# Specify which slices to show.
# Comment this out or set to zero to show all slices.
# Set to -1 to show only center slice.
# Set to -2 to show first, center and last slice.
#iv:Ge/Patient/ShowSpecificSlicesX = 1 -2
#iv:Ge/Patient/ShowSpecificSlicesY = 1 -1
#iv:Ge/Patient/ShowSpecificSlicesZ = 1 -2

# Time feature, load different CTs at different times
s:Tf/ImageName/Function = "Step"
dv:Tf/ImageName/Times = 3 10 20 30 ms
uv:Tf/ImageName/Values = 3 "synthetic_lung/lung-1" "synthetic_lung/lung-2"
→ "synthetic_lung/lung-3"
d:Tf/TimelineEnd = 120.0 ms
i:Tf/NumberOfSequentialTimes = 12
i:Tf/Verbosity = 1

s:Tf/Zoom/Function = "Step"
dv:Tf/Zoom/Times = 3 30 60 180 ms
uv:Tf/Zoom/Values = 3 1.4 3.9.
d:Tf/Zoom/RepetitionInterval = 360.0 ms

s:Tf/Pan/Function = "Step"
dv:Tf/Pan/Times = 3 30 60 180 ms
uv:Tf/Pan/Values = 3 0. .1 .1
d:Tf/Pan/RepetitionInterval = 360.0 ms

s:Tf/PanY/Function = "Step"
dv:Tf/PanY/Times = 3 30 60 180 ms
uv:Tf/PanY/Values = 3 0. -.05 -.05
d:Tf/PanY/RepetitionInterval = 360.0 ms

# Uncomment to enable scoring
s:Sc/DoseAtPhantom/Quantity = "DoseToMedium"
s:Sc/DoseAtPhantom/Component = "Patient"
b:Sc/DoseAtPhantom/OutputToConsole = "True"
s:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
s:Sc/DoseAtPhantom/OutputType = "DICOM"
u:Sc/DoseAtPhantom/DICOMOutputScaleFactor = 1.0E12
b:Sc/DoseAtPhantom/DICOMOutput32BitsPerPixel = "False"

# Uncomment to split scoring by ImageName
s:Sc/DoseAtPhantom/SplitByTimeFeature = "ImageName"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 120. MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Flat"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 0.5 cm
d:So/Example/BeamPositionCutoffY = 0.5 cm
s:So/Example/BeamAngularDistribution = "None"
i:So/Example/NumberOfHistoriesInRun = 100

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

# Graphics
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
s:Gr/ViewA/Type = "OpenGL"
DoseToCT.txt

# Demonstrates scoring in a DICOM file.
# You must unzip DICOM_Box.zip before you run this example.
# Reads in and displays a simple test DICOM that represents
# a box of water in air.
# Can optionally display patient from XiO input file.
# If Graphics is enabled, the display writes very slowly to the screen.
# This will be improved in an upcoming new release.
# Until then, one useful trick is that if you iconize the graphics window,
# then un-iconize it when drawing is done, the drawing will go much faster.

includeFile = HUtoMaterialSchneider.txt

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"
s:Ge/Patient/Parent = "World"
s:Ge/Patient/Material = "G4_WATER"
d:Ge/Patient/TransX = 0. m
d:Ge/Patient/TransY = 0. m
d:Ge/Patient/TransZ = 0. m
d:Ge/Patient/RotX = 0. deg
d:Ge/Patient/RotY = 0. deg
d:Ge/Patient/RotZ = 0. deg

# Specify which slices to show.
# Comment this out or set to zero to show all slices.
# Set to -1 to show only center slice.
# Set to -2 to show first, center and last slice.
#iv:Ge/Patient/ShowSpecificSlicesX = 1 -2
#iv:Ge/Patient/ShowSpecificSlicesY = 1 -2
#iv:Ge/Patient/ShowSpecificSlicesZ = 1 -2

# Can read either DICOM or XiO input files.
# To change input format, change which
# Ge/Patient/Type file is commented out below.
Additional needed for DICOM input:

- `:Ge/Patient/Type = "TsDicomPatient"`
- `:Ge/Patient/DicomDirectory = "DICOM_Box"`

Additional needed for XiO input:

- `:Ge/Patient/Type = "TsXiOPatient"`
- `:Ge/Patient/InputDirectory = "/"`
- `:Ge/Patient/InputFile = "XIO_Example.dat"`
- `:Ge/Patient/NumberOfVoxelsX = 25`
- `:Ge/Patient/NumberOfVoxelsY = 25`
- `:Ge/Patient/NumberOfVoxelsZ = 2 3 7`
- `:Ge/Patient/VoxelSizeX = 2.0 mm`
- `:Ge/Patient/VoxelSizeY = 2.0 mm`
- `:Ge/Patient/VoxelSizeZ = 2 5. 2.5 mm`

Dose at Phantom:

- `:Sc/DoseAtPhantom/Quantity = "DoseToMedium"
- `:Sc/DoseAtPhantom/Component = "Patient"
- `:Sc/DoseAtPhantom/OutputToConsole = "True"
- `:Sc/DoseAtPhantom/IfOutputFileAlreadyExists = "Overwrite"
- `:Sc/DoseAtPhantom/OutputType = "DICOM"
- `:Sc/DoseAtPhantom/DICOMOutputScaleFactor = 1.0E12`
- `:Sc/DoseAtPhantom/DICOMOutput32BitsPerPixel = "True"

Example:

- `:So/Example/Type = "Beam"
- `:So/Example/Component = "BeamPosition"
- `:So/Example/BeamParticle = "proton"
- `:So/Example/BeamEnergy = 30. MeV`
- `:So/Example/BeamEnergySpread = 0.757504`
- `:So/Example/BeamPositionDistribution = "Flat"
- `:So/Example/BeamPositionCutoffShape = "Ellipse"
- `:So/Example/BeamPositionCutoffX = 5. cm`
- `:So/Example/BeamPositionCutoffY = 20. cm`
- `:So/Example/BeamAngularDistribution = "None"
- `:So/Example/NumberOfHistoriesInRun = 500`

Graphics:

- `:Gr/ViewA/Type = "OpenGL"
- `:Gr/ViewA/VisibleWorlds = 1 "All"
- `:Gr/ViewA/WindowSizeX = 900`
- `:Gr/ViewA/WindowSizeY = 900`
- `:Gr/ViewA/Theta = 70 deg`
- `:Gr/ViewA/Phi = 10 deg`

Extra graphics options to handle complex voxel geometries:

- `:Gr/ShowOnlyOutlineIfVoxelCountExceeds = 210000000`
- `:Ts/ShowHistoryCountAtInterval = 100`
- `:Ts/PauseBeforeQuit = "True"`

Implant.txt:

- Demonstrates use of a DICOM file.
- You must unzip DICOM_Box.zip before you run this example.
# Reads in and displays a simple test DICOM that represents
# a box of water in air
# and uses Layered Mass Geometry to overlay a titanium sphere "implant"
# onto this DICOM geometry.
# The display writes very slowly to the screen.
# This will be improved in an upcoming new release.
# Until then, one useful trick is that if you iconize the graphics window,
# then un-inconize it when drawing is done, the drawing will go much faster.

includeFile = HUtoMaterialSchneider.txt

s:Ge/World/Material = "Vacuum"
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"
s:Ge/Patient/Parent = "World"
s:Ge/Patient/Material = "G4_WATER"
d:Ge/Patient/TransX = 0. m
d:Ge/Patient/TransY = 0. m
d:Ge/Patient/TransZ = 0. m
d:Ge/Patient/RotX = 0. deg
d:Ge/Patient/RotY = 0. deg
d:Ge/Patient/RotZ = 0. deg
s:Ge/Patient/Type = "TsDicomPatient"
s:Ge/Patient/DicomDirectory = "DICOM_Box"

# Specify which slices to show.
# Comment this out or set to zero to show all slices.
# Set to -1 to show only center slice.
# Set to -2 to show first, center and last slice.
iv:Ge/Patient/ShowSpecificSlicesX = 1 -2
iv:Ge/Patient/ShowSpecificSlicesY = 1 -2
iv:Ge/Patient/ShowSpecificSlicesZ = 1 -2

# Overlay implant in a parallel world using Layered Mass Geometry.
s:Ge/Implant/Type = "TsSphere"
s:Ge/Implant/Parent = "Patient"
b:Ge/Implant/IsParallel = "True"
s:Ge/Implant/Material = "Titanium"
d:Ge/Implant/RMin = 0. cm
d:Ge/Implant/RMax = 5.0 cm
d:Ge/Implant/SPhi = 0.0 deg
d:Ge/Implant/DPhi = 360. deg
d:Ge/Implant/STheta = 0 deg
d:Ge/Implant/DTheta = 180 deg
d:Ge/Implant/TransX = 0. cm
d:Ge/Implant/TransY = 0. cm
d:Ge/Implant/TransZ = 15. cm
d:Ge/Implant/RotX = 0. deg
d:Ge/Implant/RotY = 0.0 deg
d:Ge/Implant/RotZ = 0. deg
s:Ge/Implant/Color = "yellow"
sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
sv:Ph/Default/LayeredMassGeometryWorlds = 1 "Implant"
TOPAS Documentation, Release 3.0

ViewAbdomen.txt

# Demonstrates use of a DICOM file.
# You must unzip Abdomen.zip before you run this example.
# Reads in and displays a 3 slice scan of an abdomen.
The display writes very slowly to the screen.
This will be improved in an upcoming new release.
Until then, one useful trick is that if you iconize the graphics window,
then un-iconize it when drawing is done, the drawing will go much faster.

includeFile = HUtoMaterialSchneider.txt

s:Ge/World/Material = "Vacuum"
da:Ge/World/HLX = 2.0 m
da:Ge/World/HLY = 2.0 m
da:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"
s:Ge/Patient/Parent = "World"
s:Ge/Patient/Material = "G4_WATER"
s:Ge/Patient/Type = "TsDicomPatient"
da:Ge/Patient/TransX = 0. m
da:Ge/Patient/TransY = 0. m
da:Ge/Patient/TransZ = 0. m
da:Ge/Patient/RotX = 0. deg
da:Ge/Patient/RotY = 0. deg
da:Ge/Patient/RotZ = 0. deg
sa:Ge/Patient/DicomDirectory = "Abdomen"
sv:Ge/Patient/DicomModalityTags = 1 "CT"

Specify which slices to show.
Comment this out or set to zero to show all slices.
Set to -1 to show only center slice.
Set to -2 to show first, center and last slice.
#iv:Ge/Patient/ShowSpecificSlicesX = 1 -2
#iv:Ge/Patient/ShowSpecificSlicesY = 1 -2
#iv:Ge/Patient/ShowSpecificSlicesZ = 1 -1

sa:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
da:So/Example/BeamEnergy = 169.23 MeV
du:So/Example/BeamEnergySpread = 0.757504
sa:So/Example/BeamPositionDistribution = "Gaussian"
sa:So/Example/BeamPositionCutoffShape = "Ellipse"
da:So/Example/BeamPositionCutoffX = 10. cm
da:So/Example/BeamPositionCutoffY = 10. cm
da:So/Example/BeamPositionSpreadX = 0.65 cm
da:So/Example/BeamPositionSpreadY = 0.65 cm
sa:So/Example/BeamAngularDistribution = "Gaussian"
da:So/Example/BeamAngularCutoffX = 90. deg
da:So/Example/BeamAngularCutoffY = 90. deg
da:So/Example/BeamAngularSpreadX = 0.0032 rad
da:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

sa:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 900
d:Gr/ViewA/WindowSizeY = 900
da:Gr/ViewA/Theta = 70 deg
da:Gr/ViewA/Phi = 10 deg
i:Gr/ShowOnlyOutlineIfVoxelCountExceeds = 1000000
b:Ts/PauseBeforeQuit = "True"
# Defines a raster scanning pattern used in some demonstrations.

```plaintext
includeFile = ScanningNozzle.txt

# Time Features for Scanning Magnets

\$ Ge/BFieldX\_max = 0.15 \text{ tesla} \\
\$ Ge/BFieldX\_min = Ge/BFieldX\_max \text{ tesla} \times -1.0 \\
\$ Ge/BFieldX\_range = Ge/BFieldX\_max \text{ tesla} \times 2.0 \\

\$ Tf/BField2nd/Function = "Step" \\
\$ Tv:Tf/BField2nd/Times = 2 125.0 250.0 \text{ ms} \\
\$ Td:Tf/BField2nd/Values = 2 Td/BField2ndLeft/Value Td/BField2ndRight/Value \text{ tesla} \\

\$ Td/BField2ndLeft/Function = "Linear \text{ tesla}" \\
\$ Td/Tf/BField2ndLeft/RepetitionInterval = 125.0 \text{ ms} \\
\$ Td/Tf/BField2ndLeft/Rate = 0.0024 \text{ tesla/ms} \\
\$ Td/Tf/BField2ndLeft/StartValue = Ge/BFieldX\_min \text{ tesla} \\

\$ Td/BField2ndRight/Function = "Linear \text{ tesla}" \\
\$ Td/Tf/BField2ndRight/RepetitionInterval = 125.0 \text{ ms} \\
\$ Td/Tf/BField2ndRight/Rate = -0.0024 \text{ tesla/ms} \\
\$ Td/Tf/BField2ndRight/StartValue = Ge/BFieldX\_max \text{ tesla} \\

\$ Td/Tf/BField1st/Function = "Step" \\
\$ Td/Tf/BField1st/Times = 4 0.25 0.5 0.75 1.0 \text{ s} \\
\$ Td/Tf/BField1st/Values = 4 -0.105 -0.035 0.035 0.105 \text{ tesla} \\

\$ Ge/Dipole1/MagneticFieldStrength = Td/BField1st/Value \text{ tesla} \\
\$ Ge/Dipole2/MagneticFieldStrength = Td/BField2nd/Value \text{ tesla} 
```
# Scanning nozzle used in some demonstrations.

includeFile = ConstantsForNozzles.txt

# Ts/DumpParameters = "T"

########################################
# 1. Beam Exit Window
# 100 um thickness Mylar window
########################################
s:Ge/ExitWindow/Parent = "Gantry"
s:Ge/ExitWindow/Type = "TsCylinder"
s:Ge/ExitWindow/Material = "Mylar"
d:Ge/ExitWindow/RMin = 0.0 mm
d:Ge/ExitWindow/RMax = 50.0 mm
d:Ge/ExitWindow/HL = 0.05 mm
d:Ge/ExitWindow/SPhi = 0.0 deg
d:Ge/ExitWindow/DPhi = 360.0 deg
d:Ge/ExitWindow/TransX = 0.0 mm
d:Ge/ExitWindow/TransY = 0.0 mm
d:Ge/ExitWindow/TransZ = 299.8 cm
d:Ge/ExitWindow/RotX = 0.0 deg
d:Ge/ExitWindow/RotY = 0.0 deg
d:Ge/ExitWindow/RotZ = 0.0 deg
b:Ge/ExitWindow/Include = "TRUE"
s:Ge/ExitWindow/DrawingStyle = "Solid"

d:Ge/DistanceOf_CenterExitWindow_To_TopMon1 = -30.0 mm
d:Ge/ZPlusOf_Mon1 = Ge/ExitWindow/TransZ + Ge/DistanceOf_CenterExitWindow_To_TopMon1_ = 30.0 cm

########################################
# 2. Monitor chamber 1.
# Filled with Nitrogen Gas, 3 layers
# Thickness, position, Material
# 1.0 Gold
# 0.0 Aluminum
# -1.0 Gold
########################################
s:Ge/MonitorChamber1/Parent = "Gantry"
s:Ge/MonitorChamber1/Type = "TsCylinder"
s:Ge/MonitorChamber1/Material = "NiGas"
d:Ge/MonitorChamber1/RMin = 0.0 mm
d:Ge/MonitorChamber1/RMax = 100.0 mm
d:Ge/MonitorChamber1/HL = 15.0 mm
d:Ge/MonitorChamber1/SPhi = 0.0 deg
d:Ge/MonitorChamber1/DPhi = 360.0 deg
d:Ge/MonitorChamber1/RotX = 0.0 deg
d:Ge/MonitorChamber1/RotY = 0.0 deg
d:Ge/MonitorChamber1/RotZ = 0.0 deg
d:Ge/MonitorChamber1/TransX = 0.0 mm
d:Ge/MonitorChamber1/TransY = 0.0 mm
d:Ge/MonitorChamber1/TransZ = Ge/ZPlusOf_Mon1 + Ge/MonitorChamber1/MinusHL mm
b:Ge/MonitorChamber1/Include = "TRUE"
# Distance calculation for Pipe

\[ \text{Ge/DistanceOf_BottomMon1_To_TopPipe} = -20.0 \text{ mm} \]

\[ \text{Ge/ZMinusOf_Mon1} = \text{Ge/MonitorChamber1/TransZ} + \text{Ge/MonitorChamber1/MinusHL} \text{ mm} \]

\[ \text{Ge/ZPlusOf_Pipe} = \text{Ge/ZMinusOf_Mon1} + \text{Ge/DistanceOf_BottomMon1_To_TopPipe} \text{ mm} \]

# Quad/Dipole Field specification in length

\#Gap Quad1 Gap Quad2 Gap

\#2.0 cm + 5.0*2 + 4.0 + 5.0*2 + 2.0 = 28.0 cm
\[ \text{d:Ge/HLOf_PipeInQuads} = 14.0 \text{ cm} \]
\[ \text{d:Ge/HLOf_Quad1} = 5.0 \text{ cm} \]
\[ \text{d:Ge/HLOf_Quad2} = \text{Ge/HLOf_Quad1 cm} \]
\[ \text{d:Ge/Gap_Top_Quad1} = 2.0 \text{ cm} \]
\[ \text{d:Ge/Gap_Quad1_Quad2} = 4.0 \text{ cm} \]
\[ \text{d:Ge/Gap_Quad2_Bottom} = 2.0 \text{ cm} \]
\[ \text{d:Ge/HLxyOf_Quads} = 47.0 \text{ mm} \]
\[ \text{d:Ge/HLxyOf_Dipoles} = 77.0 \text{ mm} \]
\[ \text{d:Ge/ThicknessOf_Pipe} = 3.0 \text{ mm} \]
\[ \text{s:Ge/MaterialOf_PipeWall} = "Aluminum" \]
\[ \text{s:Ge/MaterialOf_PipeGas} = "HeGas" \]
\[ \text{s:Ge/MaterialOf_PipeGas} = "Vacuum" \]
\[ \text{d:Ge/FLOf_PipeConnector} = 10.0 \text{ cm} \]
\[ \text{d:Ge/HLOf_Dipole1} = 10.5 \text{ cm} \]
\[ \text{d:Ge/HLOf_Dipole2} = 13.0 \text{ cm} \]
\[ \text{d:Ge/DistanceOf_Coil1_Dipole1} = 6.0 \text{ cm} \]
\[ \text{d:Ge/DistanceOf_Coil2_Dipole2} = 8.0 \text{ cm} \]
\[ \text{d:Ge/DistanceOf_BottomDipoleCoil1_To_TopDipoleCoil2} = 1.0 \text{ cm} \]
\[ \text{d:Ge/DistanceOf_BottomPipe_To_TopBPM1} = -1.0 \text{ mm} \]
\[ \text{d:Ge/DistanceOf_BottomBPM1_To_PipeExtension} = -0.0 \text{ mm} \]
\[ \text{d:Ge/DistanceOf_BottomPipeExtension_To_BPM2} = -0.0 \text{ mm} \]

# Calculated parameters from input
# Don't modify following calculations
\[ \text{d:Ge/HLOf_DipoleCoil1} = \text{Ge/HLOf_Dipole1} + \text{Ge/DistanceOf_Coil1_Dipole1 cm} \]
\[ \text{d:Ge/HLOf_DipoleCoil2} = \text{Ge/HLOf_Dipole2} + \text{Ge/DistanceOf_Coil2_Dipole2 cm} \]
\[ \text{d:Ge/MinusHLOf_DipoleCoil1} = \text{Ge/HLOf_DipoleCoil1 cm} \times -1.0 \]
\[ \text{d:Ge/MinusHLOf_DipoleCoil2} = \text{Ge/HLOf_DipoleCoil2 cm} \times -1.0 \]
\[ \text{d:Ge/FLOf_DipoleCoil1} = \text{Ge/HLOf_DipoleCoil1 cm} \times 2.0 \]
\[ \text{d:Ge/FLOf_DipoleCoil2} = \text{Ge/HLOf_DipoleCoil2 cm} \times 2.0 \]
\[ \text{d:Ge/FLOf_DipoleCoil1_DipoleCoil2} = \text{Ge/FLOf_DipoleCoil1} + \text{Ge/FLOf_DipoleCoil2 cm} \]
\[ \text{d:Ge/FLOf_PipeInQuads} = 2.0 \times \text{Ge/HLOf_PipeInQuads cm} \]
\[ \text{d:Ge/FLOf_PipeConnector} = \text{Ge/FLOf_PipeConnector cm} \times 0.5 \]
\[ \text{d:Ge/MinusHLOf_PipeConnector} = \text{Ge/FLOf_PipeConnector cm} \times -0.5 \]
\[ \text{d:Ge/FLOf_PipeInDipoles} = \text{Ge/FLOf_DipoleCoil1_DipoleCoil2} + \text{Ge/DistanceOf_BottomDipoleCoil1_To_TopDipoleCoil2 cm} \]
\[ \text{d:Ge/FLOf_PipeInDipoles} = \text{Ge/FLOf_PipeInDipoles cm} \times 0.5 \]
\[ \text{d:Ge/MinusHLOf_PipeInDipoles} = \text{Ge/FLOf_PipeInDipoles cm} \times -0.5 \]
\[ \text{d:Ge/FLOf_QuadsDipoles} = \text{Ge/FLOf_PipeInDipoles} + \text{Ge/FLOf_PipeInQuads cm} \]
\[ \text{d:Ge/FLOf_Pipe} = \text{Ge/FLOf_QuadsDipoles} + \text{Ge/FLOf_PipeConnector cm} \]
\[ \text{d:Ge/HLOf_Pipe} = \text{Ge/FLOf_Pipe cm} \times 0.5 \]

# Pipe through Quads/Dipoles/
# BeamPipeInQuadsDipoles/Parent = "Gantry"
### 3. Quadrupole 1 and Quadrupole 2 in pipe

```plaintext
d:Ge/BeamPipeInQuadsDipoles/Type = "Group"
d:Ge/BeamPipeInQuadsDipoles/TransX = 0. m
d:Ge/BeamPipeInQuadsDipoles/TransY = 0. m
d:Ge/BeamPipeInQuadsDipoles/TransZ = Ge/ZPlusOf_Pipe - Ge/HLOf_Pipe mm
d:Ge/BeamPipeInQuadsDipoles/RotX = 0. deg
d:Ge/BeamPipeInQuadsDipoles/RotY = 0. deg
d:Ge/BeamPipeInQuadsDipoles/RotZ = 0. deg
d:Ge/PipeInQuads/Parent = "BeamPipeInQuadsDipoles"
d:Ge/PipeInQuads/Type = "TsBox"
d:Ge/PipeInQuads/Material = Ge/MaterialOf_PipeWall
d:Ge/PipeInQuads/HLX = Ge/HLxyOf_Quads + Ge/ThicknessOf_Pipe mm
d:Ge/PipeInQuads/HLY = Ge/HLxyOf_Quads + Ge/ThicknessOf_Pipe mm
d:Ge/PipeInQuads/HLZ = Ge/HLOf_PipeInQuads cm
d:Ge/PipeInQuads/TransX = 0.0 mm
d:Ge/PipeInQuads/TransY = 0.0 mm
d:Ge/PipeInQuads/TransZ = Ge/HLOf_Pipe - Ge/HLOf_PipeInQuads cm
d:Ge/PipeInQuads/RotX = 0.0 deg
d:Ge/PipeInQuads/RotY = 0.0 deg
d:Ge/PipeInQuads/RotZ = 0.0 deg
d:Ge/PipeInQuads/Gas/Parent = "PipeInQuads"
d:Ge/PipeInQuads/Gas/Type = "TsBox"
d:Ge/PipeInQuads/Gas/Material = Ge/MaterialOf_PipeGas
d:Ge/PipeInQuads/Gas/HLX = Ge/HLxyOf_Quads mm
d:Ge/PipeInQuads/Gas/HLY = Ge/HLxyOf_Quads mm
d:Ge/PipeInQuads/Gas/HLZ = Ge/PipeInQuads/HLZ cm
d:Ge/PipeInQuads/Gas/TransX = 0.0 mm
d:Ge/PipeInQuads/Gas/TransY = 0.0 mm
d:Ge/PipeInQuads/Gas/TransZ = 0.0 mm
d:Ge/PipeInQuads/Gas/RotX = 0.0 deg
d:Ge/PipeInQuads/Gas/RotY = 0.0 deg
d:Ge/PipeInQuads/Gas/RotZ = 0.0 deg
d:Ge/PipeInQuads/Gas/Parent = "PipeInQuads"
d:Ge/PipeInQuads/Gas/Type = "TsBox"
d:Ge/PipeInQuads/Gas/Material = Ge/MaterialOf_PipeGas
d:Ge/PipeInQuads/Gas/HLX = Ge/HLxyOf_Quads mm
d:Ge/PipeInQuads/Gas/HLY = Ge/HLxyOf_Quads mm
d:Ge/PipeInQuads/Gas/HLZ = Ge/PipeInQuads/HLZ cm
d:Ge/PipeInQuads/Gas/TransX = 0.0 mm
d:Ge/PipeInQuads/Gas/TransY = 0.0 mm
d:Ge/PipeInQuads/Gas/TransZ = 0.0 mm
d:Ge/PipeInQuads/Gas/RotX = 0.0 deg
d:Ge/PipeInQuads/Gas/RotY = 0.0 deg
d:Ge/PipeInQuads/Gas/RotZ = 0.0 deg

d:Ge/Quad1/Type = "TsBox"
d:Ge/Quad1/Parent = "PipeInQuads/Gas"
d:Ge/Quad1/Material = "parent"
d:Ge/Quad1/HLX = Ge/PipeInQuads/Gas/HLX cm
d:Ge/Quad1/HLY = Ge/PipeInQuads/Gas/HLY cm
d:Ge/Quad1/HLZ = Ge/HLOf_Quad1 cm
d:Ge/Quad1/TransX = 0.0 cm
d:Ge/Quad1/TransY = 0.0 cm
d:Ge/Quad1/Upstream = Ge/PipeInQuads/Gas/HLZ - Ge/Gap_Top_Quad1 cm
d:Ge/Quad1/TransZ = Ge/Quad1/Upstream - Ge/Quad1/HLZ cm
d:Ge/Quad1/Downstream = Ge/Quad1/TransZ - Ge/Quad1/HLZ cm
d:Ge/Quad1/RotX = 0.0 deg
d:Ge/Quad1/RotY = 0.0 deg
d:Ge/Quad1/RotZ = 0.0 deg

d:Ge/Quad1/Field = "QuadrupoleMagnet"

d:Ge/Quad1/MagneticFieldDirectionX = -1

d:Ge/Quad1/MagneticFieldDirectionY = 2.0

d:Ge/Quad1/MagneticFieldDirectionZ = 3.0

d:Ge/Quad1/MagneticFieldGradientX = 0.0 tesla/cm

d:Ge/Quad1/MagneticFieldGradientY = 0.0 tesla/cm
```

---

24.2. ScanningNozzle.txt 277
s:Ge/Quad2/Type = "TsBox"
s:Ge/Quad2/Parent = "PipeInQuads/Gas"
s:Ge/Quad2/Material = "parent"
d:Ge/Quad2/HLX = Ge/PipeInQuads/Gas/HLX cm
d:Ge/Quad2/HLZ = Ge/HLOf_Quad2 cm
d:Ge/Quad2/TransX = 0.0 cm
d:Ge/Quad2/TransY = 0.0 cm
d:Ge/Quad2/Upstream = Ge/Quad1/Downstream - Ge/Gap_Quad1_Quad2 cm
d:Ge/Quad2/TransZ = Ge/Quad2/Upstream - Ge/Quad2/HLZ cm
d:Ge/Quad2/RotX = 0.0 deg
d:Ge/Quad2/RotY = 0.0 deg
d:Ge/Quad2/RotZ = 0.0 deg
s:Ge/Quad2/Field = "QuadrupoleMagnet"
u:Ge/Quad2/MagneticFieldDirectionX = -1
u:Ge/Quad2/MagneticFieldDirectionY = 2.0
u:Ge/Quad2/MagneticFieldDirectionZ = 3.0
d:Ge/Quad2/MagneticFieldGradientX = 0.0 tesla/cm
d:Ge/Quad2/MagneticFieldGradientY = 0.0 tesla/cm
d:Ge/Quad2/MagneticFieldGradientZ = 0.0 tesla/cm
d:Ge/ZPlusPipeConnector = Ge/PipeInQuads/TransZ - Ge/HLOf_PipeInQuads cm
s:Ge/PipeConnector/Parent = "BeamPipeInQuadsDipoles"
s:Ge/PipeConnector/Type = "G4Trd"
s:Ge/PipeConnector/Material = Ge/MaterialOf_PipeWall
d:Ge/PipeConnector/HLX2 = Ge/HLxyOf_Quads + Ge/ThicknessOf_Pipe mm
d:Ge/PipeConnector/HLY2 = Ge/HLxyOf_Quads + Ge/ThicknessOf_Pipe mm
d:Ge/PipeConnector/HLX1 = Ge/HLxyOf_Dipoles + Ge/ThicknessOf_Pipe mm
d:Ge/PipeConnector/HLY1 = Ge/HLxyOf_Dipoles + Ge/ThicknessOf_Pipe mm
d:Ge/PipeConnector/HLZ = Ge/HLOf_PipeConnector mm
d:Ge/PipeConnector/TransX = 0.0 mm
d:Ge/PipeConnector/TransY = 0.0 mm
d:Ge/PipeConnector/TransZ = Ge/ZPlusPipeConnector + Ge/MinusHLOf_PipeConnector mm
d:Ge/PipeConnector/RotX = 0.0 deg
d:Ge/PipeConnector/RotY = 0.0 deg
d:Ge/PipeConnector/RotZ = 0.0 deg
b:Ge/PipeConnector/Include = "TRUE"
s:Ge/PipeConnector/Gas/Parent = "PipeConnector"
s:Ge/PipeConnector/Gas/Type = "G4Trd"
s:Ge/PipeConnector/Gas/Material = Ge/MaterialOf_PipeGas
d:Ge/PipeConnector/Gas/HLX2 = Ge/HLxyOf_Quads mm
d:Ge/PipeConnector/Gas/HLY2 = Ge/HLxyOf_Quads mm
d:Ge/PipeConnector/Gas/HLX1 = Ge/HLxyOf_Dipoles mm
d:Ge/PipeConnector/Gas/HLY1 = Ge/HLxyOf_Dipoles mm
d:Ge/PipeConnector/Gas/HLZ = Ge/PipeConnector/HLZ mm
d:Ge/PipeConnector/Gas/TransX = 0.0 mm
d:Ge/PipeConnector/Gas/TransY = 0.0 mm
d:Ge/PipeConnector/Gas/TransZ = 0.0 mm
d:Ge/PipeConnector/Gas/RotX = 0.0 deg
d:Ge/PipeConnector/Gas/RotY = 0.0 deg
d:Ge/PipeConnector/Gas/RotZ = 0.0 deg
b:Ge/PipeConnector/Gas/Include = "TRUE"
d:Ge/ZPlusPipeInDipoles = Ge/PipeConnector/TransZ + Ge/MinusHLOf_PipeConnector cm
s:Ge/PipeInDipoles/Parent = "BeamPipeInQuadsDipoles"
s:Ge/PipeInDipoles/Type = "TsBox"
**TOPAS Documentation, Release 3.0**

```plaintext
s:Ge/PipeInDipoles/Material = Ge/MaterialOf_PipeWall
d:Ge/PipeInDipoles/HLX = Ge/HLxyOf_Dipoles + Ge/ThicknessOf_Pipe mm
d:Ge/PipeInDipoles/HLY = Ge/HLxyOf_Dipoles + Ge/ThicknessOf_Pipe mm
d:Ge/PipeInDipoles/HLZ = Ge/HLOf_PipeInDipoles cm
d:Ge/PipeInDipoles/TransX = 0.0 mm
d:Ge/PipeInDipoles/TransY = 0.0 mm
d:Ge/PipeInDipoles/TransZ = Ge/ZPlusPipeInDipoles + Ge/MinusHLOf_PipeInDipoles cm
d:Ge/PipeInDipoles/RotX = 0.0 deg
d:Ge/PipeInDipoles/RotY = 0.0 deg
d:Ge/PipeInDipoles/RotZ = 0.0 deg
b:Ge/PipeInDipoles/Include = "TRUE"

s:Ge/PipeInDipoles/Gas/Parent = "PipeInDipoles"
s:Ge/PipeInDipoles/Gas/Type = "TsBox"
s:Ge/PipeInDipoles/Gas/Material = Ge/MaterialOf_PipeGas
d:Ge/PipeInDipoles/Gas/HLX = Ge/HLxyOf_Dipoles mm
d:Ge/PipeInDipoles/Gas/HLY = Ge/HLxyOf_Dipoles mm
d:Ge/PipeInDipoles/Gas/HLZ = Ge/HLOf_PipeInDipoles cm
d:Ge/PipeInDipoles/Gas/TransX = 0.0 mm
d:Ge/PipeInDipoles/Gas/TransY = 0.0 mm
d:Ge/PipeInDipoles/Gas/TransZ = 0.0 mm
d:Ge/PipeInDipoles/Gas/RotX = 0.0 deg
d:Ge/PipeInDipoles/Gas/RotY = 0.0 deg
d:Ge/PipeInDipoles/Gas/RotZ = 0.0 deg
b:Ge/PipeInDipoles/Gas/Include = "TRUE"

# 5. Dipole 1 and 2. in Pipe

s:Ge/Dipole1/Type = "TsBox"
s:Ge/Dipole1/Parent = "PipeInDipoles/Gas"
s:Ge/Dipole1/Material = "parent"
d:Ge/Dipole1/HLX = Ge/PipeInDipoles/Gas/HLX mm
d:Ge/Dipole1/HLY = Ge/PipeInDipoles/Gas/HLY mm
d:Ge/Dipole1/HLZ = Ge/HLOf_Dipole1 mm
d:Ge/Dipole1/TransX = 0.0 cm
d:Ge/Dipole1/TransY = 0.0 cm
d:Ge/Dipole1/TransZ = Ge/HLOf_PipeInDipoles + Ge/MinusHLOf_DipoleCoil1 mm
d:Ge/Dipole1/RotX = 0.0 deg
d:Ge/Dipole1/RotY = 0.0 deg
d:Ge/Dipole1/RotZ = 0.0 deg
s:Ge/Dipole1/Field = "DipoleMagnet"
u:Ge/Dipole1/MagneticFieldDirectionX = 1.0
u:Ge/Dipole1/MagneticFieldDirectionY = 0.0
u:Ge/Dipole1/MagneticFieldDirectionZ = 0.0
u:Ge/Dipole1/MagneticFieldStrength = 0.0 tesla
d:Ge/ZMinusOf_Dipole1 = Ge/Dipole1/TransZ + Ge/MinusHLOf_DipoleCoil1 mm
d:Ge/ZPlusOf_Dipole2 = Ge/ZMinusOf_Dipole1 + Ge/DistanceOf_BottomDipoleCoil1_To_TopDipoleCoil2 cm
s:Ge/Dipole2/Type = "TsBox"
s:Ge/Dipole2/Parent = "PipeInDipoles/Gas"
s:Ge/Dipole2/Material = "parent"
d:Ge/Dipole2/HLX = Ge/PipeInDipoles/Gas/HLX mm
d:Ge/Dipole2/HLY = Ge/PipeInDipoles/Gas/HLY mm
d:Ge/Dipole2/HLZ = Ge/HLOf_Dipole2 mm
d:Ge/Dipole2/TransX = 0.0 cm
```

24.2. ScanningNozzle.txt 279
\[d: Ge/Dipole2/TransY = 0.0 \text{ cm}\]
\[d: Ge/Dipole2/TransZ = Ge/ZPlusOf_Dipole2 + Ge/MinusHLOf_DipoleCoil2 \text{ mm}\]
\[d: Ge/Dipole2/RotX = 0.0 \text{ deg}\]
\[d: Ge/Dipole2/RotY = 0.0 \text{ deg}\]
\[d: Ge/Dipole2/RotZ = 0.0 \text{ deg}\]
\[s: Ge/Dipole2/Field = "DipoleMagnet"\]
\[u: Ge/Dipole2/MagneticFieldDirectionX = 0.0\]
\[u: Ge/Dipole2/MagneticFieldDirectionY = 1.0\]
\[u: Ge/Dipole2/MagneticFieldDirectionZ = 0.0\]
\[d: Ge/Dipole2/MagneticFieldStrength = 0.0 \text{ tesla}\]
\[d: Ge/ZMinusOf_Pipe = Ge/BeamPipeInQuadsDipoles/TransZ - Ge/HLOf_Pipe \text{ mm}\]
\[d: Ge/ZPlusOf_BPM1 = Ge/ZMinusOf_Pipe + Ge/DistanceOf_BottomPipe_To_TopBPM1 \text{ mm}\]

#5. Beam profile Monitor chamber at the end of pipe
HLx, HLy, HLz (9.2, 9.2, 1.8)
2 Mylar windows
2 sets of Multi-wire chamber

\[s: Ge/BeamProfileMonitor1/Parent = "Gantry"\]
\[s: Ge/BeamProfileMonitor1/Type = "TsBox"\]
\[s: Ge/BeamProfileMonitor1/Material = "NiGas"\]
\[d: Ge/BeamProfileMonitor1/HLX = 92.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/HLY = 92.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/HLZ = 22.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/MinusHL = Ge/BeamProfileMonitor1/HLZ \text{ mm} \times -1.0\]
\[d: Ge/BeamProfileMonitor1/TransX = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/TransY = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/TransZ = Ge/ZPlusOf_BPM1 + Ge/BeamProfileMonitor1/MinusHL, \text{ cm}\]
\[d: Ge/BeamProfileMonitor1/RotX = 0.0 \text{ deg}\]
\[d: Ge/BeamProfileMonitor1/RotY = 0.0 \text{ deg}\]
\[d: Ge/BeamProfileMonitor1/RotZ = 0.0 \text{ deg}\]
\[s: Ge/BeamProfileMonitor1/Include = "TRUE"\]
\[s: Ge/BeamProfileMonitor1/Window1/Parent = "BeamProfileMonitor1"\]
\[s: Ge/BeamProfileMonitor1/Window1/Type = "TsBox"\]
\[s: Ge/BeamProfileMonitor1/Window1/Material = "Mylar"\]
\[d: Ge/BeamProfileMonitor1/Window1/HLX = 90.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/HLY = 90.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/HLZ = 0.02 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/TransX = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/TransY = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/TransZ = 18.123 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window1/RotX = 0.0 \text{ deg}\]
\[d: Ge/BeamProfileMonitor1/Window1/RotY = 0.0 \text{ deg}\]
\[d: Ge/BeamProfileMonitor1/Window1/RotZ = 0.0 \text{ deg}\]
\[s: Ge/BeamProfileMonitor1/Window2/Parent = "BeamProfileMonitor1"\]
\[s: Ge/BeamProfileMonitor1/Window2/Type = "TsBox"\]
\[s: Ge/BeamProfileMonitor1/Window2/Material = "Mylar"\]
\[d: Ge/BeamProfileMonitor1/Window2/HLX = 90.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window2/HLY = 90.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window2/HLZ = 0.02 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window2/TransX = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window2/TransY = 0.0 \text{ mm}\]
\[d: Ge/BeamProfileMonitor1/Window2/TransZ = -18.123 \text{ mm}\]
d:Ge/ZMinusOf_BPM1 = Ge/BeamProfileMonitor1/TransZ + Ge/BeamProfileMonitor1/MinusHL mm

d:Ge/ZPlusOf_PipeExtension = Ge/ZMinusOf_BPM1 + Ge/DistanceOf_BottomBPM1_To_PipeExtension mm

d:Ge/FLOf_PipeExtension = 80.0 cm

d:Ge/HLOf_PipeExtension = Ge/FLOf_PipeExtension cm * 0.5

d:Ge/MinusHLOf_PipeExtension = Ge/FLOf_PipeExtension cm * -0.5

# 6. Beam pipe extension
# HLx, HLy, (9.2, 9.2)
# HLx, HLy, HLz (1
# 2 Mylar windows
# 2 sets of Multi-wire chamber

s:Ge/PipeExtension/Parent = "Gantry"
s:Ge/PipeExtension/Type = "G4Trd"
s:Ge/PipeExtension/Material = "Aluminum"
d:Ge/PipeExtension/HLX2 = 92.0 mm
d:Ge/PipeExtension/HLY2 = 92.0 mm
d:Ge/PipeExtension/HLX1 = 122.0 mm
d:Ge/PipeExtension/HLY1 = 122.0 mm
d:Ge/PipeExtension/HLZ = Ge/HLOf_PipeExtension mm
d:Ge/PipeExtension/TransX = 0.0 mm
d:Ge/PipeExtension/TransY = 0.0 mm
d:Ge/PipeExtension/TransZ = Ge/PipeExtension/TransZ + Ge/MinusHLOf_PipeExtension mm
d:Ge/PipeExtension/RotX = 0.0 deg
d:Ge/PipeExtension/RotY = 0.0 deg
d:Ge/PipeExtension/RotZ = 0.0 deg
b:Ge/PipeExtension/Include = "TRUE"

s:Ge/PipeExtension/Gas/Parent = "PipeExtension"
s:Ge/PipeExtension/Gas/Type = "G4Trd"
s:Ge/PipeExtension/Gas/Material = Ge/PipeInQuads/Gas/Material
d:Ge/PipeExtension/Gas/HLX2 = 90.0 mm
d:Ge/PipeExtension/Gas/HLY2 = 90.0 mm
d:Ge/PipeExtension/Gas/HLX1 = 120.0 mm
d:Ge/PipeExtension/Gas/HLY1 = 120.0 mm
d:Ge/PipeExtension/Gas/HLZ = Ge/PipeExtension/HLZ mm
d:Ge/PipeExtension/Gas/TransX = 0.0 mm
d:Ge/PipeExtension/Gas/TransY = 0.0 mm
d:Ge/PipeExtension/Gas/TransZ = 0.0 mm
d:Ge/PipeExtension/Gas/RotX = 0.0 deg
d:Ge/PipeExtension/Gas/RotY = 0.0 deg
d:Ge/PipeExtension/Gas/RotZ = 0.0 deg
b:Ge/PipeExtension/Gas/Include = "TRUE"

s:Ge/PipeExtension/ZPlusOf_PipeExtension = Ge/PipeExtension/TransZ + Ge/MinusHLOf_PipeExtension mm
## 6. Beam profile monitor 2

- 2 Mylar windows
- 2 sets of Multi-wire chamber

```plaintext
# TOPAS Documentation, Release 3.0

d Ge/ZPlusOf_BPM2 = Ge/ZMinusOf_PipeExtension + Ge/DistanceOf_BottomPipeExtension_To_BPM2 mm

#6. Beam profile monitor 2
# 2 Mylar windows
# 2 sets of Multi-wire chamber

s Ge/BeamProfileMonitor2/Parent = "Gantry"
s Ge/BeamProfileMonitor2/Type = "TsBox"
s Ge/BeamProfileMonitor2/Material = "NiGas"
d Ge/BeamProfileMonitor2/HLX = 152.0 mm
d Ge/BeamProfileMonitor2/HLY = 152.0 mm
d Ge/BeamProfileMonitor2/HLZ = 25.0 mm
d Ge/BeamProfileMonitor2/MinusHL = Ge/BeamProfileMonitor2/HLZ mm * -1.0
d Ge/BeamProfileMonitor2/TransX = 0.0 mm
d Ge/BeamProfileMonitor2/TransY = 0.0 mm
d Ge/BeamProfileMonitor2/TransZ = Ge/ZPlusOf_BPM2 + Ge/BeamProfileMonitor2/MinusHL mm

d Ge/BeamProfileMonitor2/RotX = 0.0 deg
d Ge/BeamProfileMonitor2/RotY = 0.0 deg
d Ge/BeamProfileMonitor2/RotZ = 0.0 deg
b Ge/BeamProfileMonitor2/Include = "TRUE"

s Ge/BeamProfileMonitor2/Window1/Parent = "BeamProfileMonitor2"
s Ge/BeamProfileMonitor2/Window1/Type = "TsBox"
s Ge/BeamProfileMonitor2/Window1/Material = "Mylar"
d Ge/BeamProfileMonitor2/Window1/HLX = 150.0 mm
d Ge/BeamProfileMonitor2/Window1/HLY = 150.0 mm
d Ge/BeamProfileMonitor2/Window1/HLZ = 0.01 mm
d Ge/BeamProfileMonitor2/Window1/TransX = 0.0 mm
d Ge/BeamProfileMonitor2/Window1/TransY = 0.0 mm
d Ge/BeamProfileMonitor2/Window1/TransZ = 18.123 mm
d Ge/BeamProfileMonitor2/Window1/RotX = 0.0 deg
d Ge/BeamProfileMonitor2/Window1/RotY = 0.0 deg
d Ge/BeamProfileMonitor2/Window1/RotZ = 0.0 deg

s Ge/BeamProfileMonitor2/Window2/Parent = "BeamProfileMonitor2"
s Ge/BeamProfileMonitor2/Window2/Type = "TsBox"
s Ge/BeamProfileMonitor2/Window2/Material = "Mylar"
d Ge/BeamProfileMonitor2/Window2/HLX = 150.0 mm
d Ge/BeamProfileMonitor2/Window2/HLY = 150.0 mm
d Ge/BeamProfileMonitor2/Window2/HLZ = 0.01 mm
d Ge/BeamProfileMonitor2/Window2/TransX = 0.0 mm
d Ge/BeamProfileMonitor2/Window2/TransY = 0.0 mm
d Ge/BeamProfileMonitor2/Window2/TransZ = -18.123 mm
d Ge/BeamProfileMonitor2/Window2/RotX = 0.0 deg
d Ge/BeamProfileMonitor2/Window2/RotY = 0.0 deg
d Ge/BeamProfileMonitor2/Window2/RotZ = 0.0 deg

s Ge/MWC_In_BPM2/Parent = "BeamProfileMonitor2"
s Ge/MWC_In_BPM2/Type = "TsMultiWireChamber"
s Ge/MWC_In_BPM2/Material = "parent"
d Ge/MWC_In_BPM2/HLX = 148.0 mm
d Ge/MWC_In_BPM2/HLY = 148.0 mm
d Ge/MWC_In_BPM2/HLZ = 13.0 mm
d Ge/MWC_In_BPM2/TransX = 0.0 mm
d Ge/MWC_In_BPM2/TransY = 0.0 mm
```
### ScanningStationaryTarget.txt

* Use scanning nozzle to paint stationary target

```plaintext
includeFile = RasterScanningPattern.txt TargetInWaterPhantom.txt Viewer.txt

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
```
b:Ts/PauseBeforeQuit = "False"

i:So/Example/NumberOfHistoriesInRun = 5
d:So/Example/BeamEnergy = 145.00 MeV

d:Tf/TimelineEnd = 1 s
i:Tf/NumberOfSequentialTimes = 100

ScanningTargetMovingHorizontal.txt

# Use scanning nozzle to paint target that is moving horizontally

includeFile = RasterScanningPattern.txt TargetInWaterPhantom.txt Viewer.txt

i:Tf/Verbosity = 1

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

d:Ge/X_max = 1.5 cm
d:Ge/X_min = Ge/X_max cm * -1.0
d:Ge/X_range = Ge/X_max cm * 2.0

# 1. Motion for Target

s:Tf/HorizontalMoveSine/Function = "Sine"
d:Tf/HorizontalMoveSine/RepetitionInterval = 1.0 s
d:Tf/HorizontalMoveSine/Rate = 0.36 deg/ms
d:Tf/HorizontalMoveSine/StartValue = 0.0 deg

d:Tf/HorizontalMove/Value = 2.0 cm * Tf/HorizontalMoveSine/Value
d:Tf/HorizontalScan/Value = 0.15 tesla * Tf/HorizontalMoveSine/Value

# 2. Compensated BField

d:Tf/BField2ndTrackingField/Value = Tf/HorizontalScan/Value + Tf/BField2nd/Value tesla

i:Tf/NumberOfSequentialTimes = 100
d:Tf/TimelineEnd = 1 s

d:Ge/PlaneAtR80/TransX = Tf/HorizontalMove/Value cm
d:Ge/Dipole2/MagneticFieldStrength = Tf/BField2ndTrackingField/Value tesla

ScanningTargetMovingInDepth.txt

# Use scanning nozzle to paint target this is moving in depth

includeFile = RasterScanningPattern.txt TargetInWaterPhantom.txt Viewer.txt

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

b:Ts/PauseBeforeQuit = "False"

i:So/Example/NumberOfHistoriesInRun = 10
d:Tf/TimelineEnd = 1 s
#1. Target position and Energy varies as a sinusoidal

- \( \text{Tf/NumberOfSequentialTimes} = 100 \)
- \( \text{Tf/VerticalMoveCosine/Function} = \text{"Cosine"} \)
- \( \text{Tf/VerticalMoveCosine/RepetitionInterval} = 1.0 \text{ s} \)
- \( \text{Tf/VerticalMoveCosine/Rate} = 0.36 \text{ deg/ms} \)
- \( \text{Tf/VerticalMoveCosine/StartValue} = 0.0 \text{ deg} \)

#2. Tracking target

- \( \text{Tf/VerticalMoveD_Cosine/Value} = 18.6 \text{ mm} \times \text{Tf/VerticalMoveCosine/Value} \)
- \( \text{Tf/VerticalMoveR80/Value} = -166.142 \text{ mm} + \text{Tf/VerticalMoveD_Cosine/Value} \)

#3. Beam energy changing for tracking target

- \( \text{Tf/VerticalMoveE_Cosine/Value} = -10.0 \text{ MeV} \times \text{Tf/VerticalMoveCosine/Value} \)
- \( \text{Tf/VerticalMoveEnergy/Value} = 155.0 \text{ MeV} + \text{Tf/VerticalMoveE_Cosine/Value} \)
- \( \text{So/Example/BeamEnergy} = \text{Tf/VerticalMoveEnergy/Value} \text{ MeV} \)
- \( \text{Ge/R80} = \text{Tf/VerticalMoveR80/Value} \text{ cm} \)

#Fixed target for visual guidance

- \( \text{Ge/R80Fix} = -140.50 \text{ mm} \)
- \( \text{Ge/PlaneAtR80Fix/Parent} = \text{"WaterPhantom"} \)
- \( \text{Ge/PlaneAtR80Fix/Type} = \text{"TsBox"} \)
- \( \text{Ge/PlaneAtR80Fix/Material} = \text{"G4 \_WATER"} \)
- \( \text{Ge/PlaneAtR80Fix/HLX} = 5.0 \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/HLY} = 5.0 \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/HLZ} = 0.01 \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/TransX} = 0. \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/TransY} = 0. \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/TransZ} = \text{Ge/WaterPhantom/HLZ} + \text{Ge/R80Fix} \text{ cm} \)
- \( \text{Ge/PlaneAtR80Fix/RotX} = 0.0 \text{ deg} \)
- \( \text{Ge/PlaneAtR80Fix/RotY} = 0.0 \text{ deg} \)
- \( \text{Ge/PlaneAtR80Fix/RotZ} = 0.0 \text{ deg} \)
- \( \text{Ge/PlaneAtR80Fix/Color} = \text{"green"} \)
- \( \text{i/PlaneAtR80Fix/XBins} = 1 \)
- \( \text{i/PlaneAtR80Fix/YBins} = 1 \)
- \( \text{i/PlaneAtR80Fix/ZBins} = 1 \)

#Scattering nozzle used in some demonstrations.

`includeFile = ConstantsForNozzles.txt`

`ScatteringNozzle.txt`
---

### 2. Monitor chamber 1.

- **Filled with Nitrogen Gas, 3 layers**
- **Thickness, position, Material**
  - 1.0 Gold
  - 0.0 Aluminum
  - -1.0 Gold

---

```plaintext
#2:Ge/MonitorChamber1/Parent = "Gantry"
#2:Ge/MonitorChamber1/Type = "TsCylinder"
#2:Ge/MonitorChamber1/Material = "NiGas"
#2:Ge/MonitorChamber1/RMin = 0.0 mm
#2:Ge/MonitorChamber1/RMax = 100.0 mm
#2:Ge/MonitorChamber1/HL = 15.0 mm
#2:Ge/MonitorChamber1/SPhi = 0.0 deg
#2:Ge/MonitorChamber1/DPhi = 360.0 deg
#2:Ge/MonitorChamber1/RotX = 0.0 deg
#2:Ge/MonitorChamber1/RotY = 0.0 deg
#2:Ge/MonitorChamber1/RotZ = 0.0 deg
#2:Ge/MonitorChamber1/TransX = 0.0 mm
#2:Ge/MonitorChamber1/TransY = 0.0 mm
#2:Ge/MonitorChamber1/MinusHL = Ge/MonitorChamber1/HL mm * -1.0
#2:Ge/MonitorChamber1/TransZ = Ge/ZPlusOf_Mon1 + Ge/MonitorChamber1/MinusHL mm
#2:Ge/MonitorChamber1/Include = "TRUE"
#2:Ge/MonitorChamber1/Layer1/Parent = "MonitorChamber1"
#2:Ge/MonitorChamber1/Layer1/Type = "TsCylinder"
#2:Ge/MonitorChamber1/Layer1/Material = "Gold"
#2:Ge/MonitorChamber1/Layer1/RMin = 0.0 mm
#2:Ge/MonitorChamber1/Layer1/RMax = 90.0 mm
#2:Ge/MonitorChamber1/Layer1/HL = 0.002 mm
#2:Ge/MonitorChamber1/Layer1/SPhi = 0.0 deg
#2:Ge/MonitorChamber1/Layer1/DPhi = 360.0 deg
#2:Ge/MonitorChamber1/Layer1/RotX = 0.0 deg
#2:Ge/MonitorChamber1/Layer1/RotY = 0.0 deg
#2:Ge/MonitorChamber1/Layer1/RotZ = 0.0 deg
#2:Ge/MonitorChamber1/Layer1/TransX = 0.0 mm
#2:Ge/MonitorChamber1/Layer1/TransY = 0.0 mm
```

---
### First Scatterer:

#### First Scatterer:

```
# Center for rotation of scatterers:
d:Ge/Scatterer1/RotationCenterY = 5.0 cm
# Defining the in and out of beam rotation angle
d:Ge/Scatterer1/RotZ_InBeam = 180.0 deg
d:Ge/Scatterer1/RotZ_OutOfBeam = 0.0 deg
```
# Lollipop 1:
`s:Ge/Scatterer1/L1/Type = "Group"
s:Ge/Scatterer1/L1/Parent = "Scatterer1"
d:Ge/Scatterer1/L1/TransX = 0.0 cm
d:Ge/Scatterer1/L1/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L1/TransZ = 0.0 cm
d:Ge/Scatterer1/L1/RotX = 0. deg
d:Ge/Scatterer1/L1/RotY = 0. deg
d:Ge/Scatterer1/L1/RotZ = Ge/Scatterer1/Lollipop1 deg

# L1 Lead Foil:
`s:Ge/Scatterer1/L1/LeadFoil/Parent = "Scatterer1/L1"
s:Ge/Scatterer1/L1/LeadFoil/Type = "TsBox"
s:Ge/Scatterer1/L1/LeadFoil/Material = "Lead"
d:Ge/Scatterer1/L1/LeadFoil/HLX = 2.15 cm
d:Ge/Scatterer1/L1/LeadFoil/HLY = 2.15 cm
d:Ge/Scatterer1/L1/LeadFoil/HLZ = 0.01 cm
d:Ge/Scatterer1/L1/LeadFoil/TransX = 0.0 cm
d:Ge/Scatterer1/L1/LeadFoil/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L1/LeadFoil/TransZ = 1.0 cm
d:Ge/Scatterer1/L1/LeadFoil/RotX = 0. deg
d:Ge/Scatterer1/L1/LeadFoil/RotY = 0. deg
d:Ge/Scatterer1/L1/LeadFoil/RotZ = 0. deg

# Lollipop 2:
`s:Ge/Scatterer1/L2/Type = "Group"
s:Ge/Scatterer1/L2/Parent = "Scatterer1"
d:Ge/Scatterer1/L2/TransX = 0.0 cm
d:Ge/Scatterer1/L2/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L2/TransZ = 0.0 cm
d:Ge/Scatterer1/L2/RotX = 0. deg
d:Ge/Scatterer1/L2/RotY = 0. deg
d:Ge/Scatterer1/L2/RotZ = Ge/Scatterer1/Lollipop2 deg

# L2 Lead Foil:
`s:Ge/Scatterer1/L2/LexanFoil/Parent = "Scatterer1/L2"
s:Ge/Scatterer1/L2/LexanFoil/Type = "TsBox"
s:Ge/Scatterer1/L2/LexanFoil/Material = "Lexan"
d:Ge/Scatterer1/L2/LexanFoil/HLX = 2.5 cm
d:Ge/Scatterer1/L2/LexanFoil/HLY = 2.5 cm
d:Ge/Scatterer1/L2/LexanFoil/HLZ = 0.2 cm
d:Ge/Scatterer1/L2/LexanFoil/TransX = 0.0 cm
d:Ge/Scatterer1/L2/LexanFoil/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L2/LexanFoil/TransZ = 0.0 cm
d:Ge/Scatterer1/L2/LexanFoil/RotX = 0. deg
d:Ge/Scatterer1/L2/LexanFoil/RotY = 0. deg
d:Ge/Scatterer1/L2/LexanFoil/RotZ = 0. deg

# Lollipop 3:
`s:Ge/Scatterer1/L3/Type = "Group"
s:Ge/Scatterer1/L3/Parent = "Scatterer1"
d:Ge/Scatterer1/L3/TransX = 0.0 cm
d:Ge/Scatterer1/L3/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L3/TransZ = 0.0 cm
d:Ge/Scatterer1/L3/RotX = 0. deg
d:Ge/Scatterer1/L3/RotY = 0. deg
d:Ge/Scatterer1/L3/RotZ - Ge/Scatterer1/Lollipop2 deg

# L2 Lead Foil:
s:Ge/Scatterer1/L3/LeadFoil/Parent = "Scatterer1/L3"
s:Ge/Scatterer1/L3/LeadFoil/Type = "TsBox"
s:Ge/Scatterer1/L3/LeadFoil/Material = "Lead"
d:Ge/Scatterer1/L3/LeadFoil/HLX = 2.5 cm
d:Ge/Scatterer1/L3/LeadFoil/HLY = 2.5 cm
d:Ge/Scatterer1/L3/LeadFoil/HLZ = 0.02 cm
d:Ge/Scatterer1/L3/LeadFoil/TransX = 0.0 cm
d:Ge/Scatterer1/L3/LeadFoil/TransY = Ge/Scatterer1/RotationCenterY cm
d:Ge/Scatterer1/L3/LeadFoil/TransZ = -1.0 cm
d:Ge/Scatterer1/L3/LeadFoil/RotX = 0. deg
d:Ge/Scatterer1/L3/LeadFoil/RotY = 0. deg
d:Ge/Scatterer1/L3/LeadFoil/RotZ = 0. deg

########################################
#. Range modulator
# (Copied from SpecialComponents/RangeModulator.txt
########################################
s:Ge/RMW_1/Type = "TsRangeModulator"
s:Ge/RMW_1/Material = "Parent"
s:Ge/RMW_1/Parent = "Gantry"
d:Ge/RMW_1/TransX = 10.0 cm
d:Ge/RMW_1/TransY = 0.0 cm
d:Ge/RMW_1/TransZ = 250.0 cm
d:Ge/RMW_1/RotX = 0.0 deg
d:Ge/RMW_1/RotY = 0.0 deg
d:Ge/RMW_1/RotZ = 0.0 deg
b:Ge/RMW_1/Invisible = "TRUE"
b:Ge/RMW_1/PrintInformation = "True"
d:Ge/RMW_1/HeightOfUpper = 150 mm
d:Ge/RMW_1/HeightOfMiddle = 1.0 mm
d:Ge/RMW_1/HeightOfLower = 9.0 mm
d:Ge/RMW_1/Shell/Rin = 15.0 cm
d:Ge/RMW_1/Shell/Rout = 15.5 cm
s:Ge/RMW_1/Shell/Material = "Aluminum"
s:Ge/RMW_1/Shell/Color = "grey"
s:Ge/RMW_1/Shell/DrawingStyle = "Solid"
i:Ge/RMW_1/Shell/VisSegsPerCircle = 360
d:Ge/RMW_1/Hub/Rin = 6.0 cm
d:Ge/RMW_1/Hub/Rout = 7.0 cm
s:Ge/RMW_1/Hub/Material = "Aluminum"
s:Ge/RMW_1/Hub/Color = "grey"
s:Ge/RMW_1/Hub/DrawingStyle = "Solid"
i:Ge/RMW_1/Hub/VisSegsPerCircle = 360

#Upper tracks
dv:Ge/RMW_1/Upper/RadialDivisions = 1 11.0 cm
s:Ge/RMW_1/Upper/Track1/Pattern = "LexanBlockT1"
s:Ge/RMW_1/Upper/Track2/Pattern = "NULL"
#Middle tracks
dv:Ge/RMW_1/Middle/RadialDivisions = 1 11.0 cm
s:Ge/RMW_1/Middle/Track1/Pattern = "InterfaceDisk"
## Ge/RMW_1/Middle/Track2/Pattern = "HoleTrackDisk"

# Lower tracks

dv: Ge/RMW_1/Lower/Track1/Pattern = "LeadBlockT1"

dv: Ge/RMW_1/Lower/Track2/Pattern = "NULL"

# A track pattern: 14 blocks of Lexan

dv: Ge/LexanBlockT1/Offset = 0.0 deg

dv: Ge/LexanBlockT1/Angles = 14
  3.00 115.00 146.50 173.2 195.07
  216.15 230.14 243.00 255.5 270.60
  282.20 294.60 306.20 324.00 deg
dv: Ge/LexanBlockT1/Heights = 14
  77.0 82.0 87.0 92.15 95.0
  100.4 106.0 110.2 115.3 119.5
  124.0 128.8 132.0 60.0 mm
sv: Ge/LexanBlockT1/Materials = 14
  "Lexan" "Lexan" "Lexan" "Lexan" "Lexan"
  "Lexan" "Lexan" "Lexan" "Lexan"
  "Lexan" "Lexan" "Lexan" "Brass"

# A track pattern of single block

dv: Ge/InterfaceDisk/Angles = 1 0.0 deg

dv: Ge/InterfaceDisk/Heights = 1 1.0 mm
sv: Ge/InterfaceDisk/Materials = 1 "Aluminum"

# A track pattern of two blocks but one block is a hole

dv: Ge/HoleTrackDisk/Angles = 2 90.0 110.0 deg

dv: Ge/HoleTrackDisk/Heights = 2 0.0 1.0 mm
sv: Ge/HoleTrackDisk/Materials = 2 "NULL" "Aluminum"

# A track pattern of 8 Lead blocks

dv: Ge/LeadBlockT1/Angles = 9
  5.00 115.00 146.50 173.2 195.07
  216.15 230.14 243.00 255.5 deg

dv: Ge/LeadBlockT1/Heights = 9
  0.890 0.75 0.60 0.52 0.40
  0.30 0.16 0.070 0.0 mm
sv: Ge/LeadBlockT1/Materials = 9
  "Lead" "Lead" "Lead" "Lead" "Lead"
  "Lead" "Lead" "Lead" "NULL"

# Second Scatterer:

---

# Scatterer 2 Brass Box (Mother Volume)

s: Ge/Scatterer2/BrassBox/Parent = "Scatterer2"
```plaintext
# Scatterer Group:
:Ge/Scatterer2/BrassBox/Type = "TsBox"
:Ge/Scatterer2/BrassBox/Material = "Brass"
:Ge/Scatterer2/BrassBox/HLX = 30.0 cm
:Ge/Scatterer2/BrassBox/HLY = 30.0 cm
:Ge/Scatterer2/BrassBox/HLZ = 3.0 cm
:Ge/Scatterer2/BrassBox/TransX = 0.0 cm
:Ge/Scatterer2/BrassBox/TransY = 0.0 cm
:Ge/Scatterer2/BrassBox/TransZ = 0.0 cm
:Ge/Scatterer2/BrassBox/RotX = 0.0 deg
:Ge/Scatterer2/BrassBox/RotY = 0.0 deg
:Ge/Scatterer2/BrassBox/RotZ = 0.0 deg

# Scatter Holder Group:
:Ge/Scatterer2/Holder/Type = "Group"
:Ge/Scatterer2/Holder/Parent = "Scatterer2/BrassBox"
:Ge/Scatterer2/Holder/TransX = 10.0 cm
:Ge/Scatterer2/Holder/TransY = 10.0 cm
:Ge/Scatterer2/Holder/TransZ = 0.0 cm
:Ge/Scatterer2/Holder/RotX = 0.0 deg
:Ge/Scatterer2/Holder/RotY = 0.0 deg
:Ge/Scatterer2/Holder/RotZ = 0.0 deg

# Setting up the rotations for selecting a scatterer, this can be made Gantry
-specific:
:Ge/Scatterer2/RotZForS1 = 180.0 deg
:Ge/Scatterer2/RotZForS2 = 0.0 deg

# Scatter Hole 1
:Ge/Scatterer2/Hole1/Type = "Group"
:Ge/Scatterer2/Hole1/Parent = "Scatterer2/Holder"
:Ge/Scatterer2/Hole1/TransX = 10.0 cm
:Ge/Scatterer2/Hole1/TransY = 10.0 cm
:Ge/Scatterer2/Hole1/TransZ = 0.0 cm
:Ge/Scatterer2/Hole1/RotX = 0.0 deg
:Ge/Scatterer2/Hole1/RotY = 0.0 deg
:Ge/Scatterer2/Hole1/RotZ = 0.0 deg

# Scatter Hole 2
:Ge/Scatterer2/Hole2/Type = "Group"
:Ge/Scatterer2/Hole2/Parent = "Scatterer2/Holder"
:Ge/Scatterer2/Hole2/TransX = -10.0 cm
:Ge/Scatterer2/Hole2/TransY = -10.0 cm
:Ge/Scatterer2/Hole2/TransZ = 0.0 cm
:Ge/Scatterer2/Hole2/RotX = 0.0 deg
:Ge/Scatterer2/Hole2/RotY = 0.0 deg
:Ge/Scatterer2/Hole2/RotZ = 0.0 deg

# Scatterer 1 (no scattering)
# Air Tube: (no scattering, for pencil beam etc)
:Ge/Scatterer2/S1/Parent = "Scatterer2/Hole1"
:Ge/Scatterer2/S1/Type = "TsCylinder"
:Ge/Scatterer2/S1/Material = "World"
:Ge/Scatterer2/S1/RMin = 0.0 cm
:Ge/Scatterer2/S1/RMax = 5.0 cm
:Ge/Scatterer2/S1/HL = 3.0 cm
:Ge/Scatterer2/S1/TransX = 0.0 cm
:Ge/Scatterer2/S1/TransY = 0.0 cm
:Ge/Scatterer2/S1/TransZ = 0.0 cm
```

Chapter 24. Nozzle
TOPAS Documentation, Release 3.0

# Scatterer 2 (example second scatterer)
# Air Hole for Scatterer 2:
s:Ge/Scatterer2/S2/Parent = "Scatterer2/Hole2"
s:Ge/Scatterer2/S2/Type = "TsCylinder"
s:Ge/Scatterer2/S2/Material = "World"
d:Ge/Scatterer2/S2/RMin = 0.0 cm
d:Ge/Scatterer2/S2/RMax = 5.0 cm
d:Ge/Scatterer2/S2/HL = 3.0 cm
d:Ge/Scatterer2/S2/TransX = 0.0 cm
d:Ge/Scatterer2/S2/TransY = 0.0 cm
d:Ge/Scatterer2/S2/TransZ = 0.0 cm
d:Ge/Scatterer2/S2/RotX = 0.0 deg
d:Ge/Scatterer2/S2/RotY = 0.0 deg
d:Ge/Scatterer2/S2/RotZ = 0.0 deg
d:Ge/Scatterer2/S2/SPhi = 0.0 deg
d:Ge/Scatterer2/S2/DPhi = 360.0 deg

# Lead Polycone Scatterer2:
s:Ge/Scatterer2/S2/LeadPolycone/Parent = "Scatterer2/S2"
s:Ge/Scatterer2/S2/LeadPolycone/Type = "G4HPolycone"
s:Ge/Scatterer2/S2/LeadPolycone/Material = "Lead"
dv:Ge/Scatterer2/S2/LeadPolycone/RInner = 31 4.0 3.9 3.8 3.7 3.6 3.5 3.4 3.3 3.2 3.1 3.0 2.9 2.8 2.7 2.6 2.5 2.4 2.3 2.2 2.1 2.0 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0 cm
dv:Ge/Scatterer2/S2/LeadPolycone/ROuter = 31 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 cm
dv:Ge/Scatterer2/S2/LeadPolycone/Z = 31 1.0 1.0 1.0 1.0 1.0 1.0 0.98 0.96 0.94 0.92 0.9 0.88 0.86 0.84 0.82 0.8 0.78 0.76 0.74 0.72 0.7 0.68 0.66 0.64 0.62 0.6 0.58 0.56 0.54 0.52 0.5 0.48 0.46 0.44 0.42 0.4 0.38 0.36 0.34 0.32 0.3 0.28 0.26 0.24 0.22 0.2 cm

d:Ge/Scatterer2/S2/LeadPolycone/TransX = 0.0 cm
d:Ge/Scatterer2/S2/LeadPolycone/TransY = 0.0 cm
d:Ge/Scatterer2/S2/LeadPolycone/TransZ = 0.0 cm
d:Ge/Scatterer2/S2/LeadPolycone/RotX = 0.0 deg
d:Ge/Scatterer2/S2/LeadPolycone/RotY = 0.0 deg
d:Ge/Scatterer2/S2/LeadPolycone/RotZ = 0.0 deg
d:Ge/Scatterer2/S2/LeadPolycone/PhiStart = 0.0 deg
d:Ge/Scatterer2/S2/LeadPolycone/PhiTotal = 360.0 deg

# Lexan Polycone Scatterer2:
s:Ge/Scatterer2/S2/LexanPolycone/Parent = "Scatterer2/S2"
s:Ge/Scatterer2/S2/LexanPolycone/Type = "G4HPolycone"
s:Ge/Scatterer2/S2/LexanPolycone/Material = "Lexan"
dv:Ge/Scatterer2/S2/LexanPolycone/RInner = 41 4.0 3.9 3.8 3.7 3.6 3.5 3.4 3.3 3.2 3.1 3.0 2.9 2.8 2.7 2.6 2.5 2.4 2.3 2.2 2.1 2.0 1.9 1.8 1.7 1.6 1.5 1.4 1.3 1.2 1.1 1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0 cm
dv:Ge/Scatterer2/S2/LexanPolycone/ROuter = 41 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 4.0 cm
dv:Ge/Scatterer2/S2/LexanPolycone/Z = 41 1.0 1.0 1.0 1.0 1.0 1.0 0.98 0.96 0.94 0.92 0.9 0.88 0.86 0.84 0.82 0.8 0.78 0.76 0.74 0.72 0.7 0.68 0.66 0.64 0.62 0.6 0.58 0.56 0.54 0.52 0.5 0.48 0.46 0.44 0.42 0.4 0.38 0.36 0.34 0.32 0.3 0.28 0.26 0.24 0.22 0.2 cm

d:Ge/Scatterer2/S2/LexanPolycone/TransX = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/TransY = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/TransZ = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/RotX = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/RotY = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/RotZ = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/PhiStart = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/PhiTotal = 360.0 deg
```plaintext
d:Ge/Scatterer2/S2/LexanPolycone/TransX = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/TransY = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/TransZ = 0.0 cm
d:Ge/Scatterer2/S2/LexanPolycone/RotX = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/RotY = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/RotZ = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/PhiStart = 0.0 deg
d:Ge/Scatterer2/S2/LexanPolycone/PhiTotal = 360.0 deg

# Snout:

s:Ge/Snout/Type = "Group"
s:Ge/Snout/Parent = "Gantry"
d:Ge/Snout/TransX = 0. m
d:Ge/Snout/TransY = 0. m
d:Ge/Snout/TransZ = 50. cm
d:Ge/Snout/RotX = 0. deg
d:Ge/Snout/RotY = 0. deg
d:Ge/Snout/RotZ = 0. deg
s:Ge/Snout/Message = "Constructing Snout"

# Snout: Block

s:Ge/Snout/Block/Parent = "Snout"
s:Ge/Snout/Block/Type = "TsBox"
s:Ge/Snout/Block/Material = "Brass"
d:Ge/Snout/Block/HLX = 20.0 cm
d:Ge/Snout/Block/HLY = 20.0 cm
d:Ge/Snout/Block/HLZ = 3.5 cm
d:Ge/Snout/Block/TransX = 0. cm
d:Ge/Snout/Block/TransY = 0. cm
d:Ge/Snout/Block/TransZ = 0. cm
d:Ge/Snout/Block/RotX = 0. deg
d:Ge/Snout/Block/RotY = 0. deg
d:Ge/Snout/Block/RotZ = 0. deg
s:Ge/Snout/Hole/Parent = "Snout/Block"
s:Ge/Snout/Hole/Type = "TsCylinder"
s:Ge/Snout/Hole/Material = "Brass"
d:Ge/Snout/Hole/RMin = 0.0 cm
d:Ge/Snout/Hole/RMax = 4.5 cm
d:Ge/Snout/Hole/HL = 3.5 cm
d:Ge/Snout/Hole/TransX = 0.0 cm
d:Ge/Snout/Hole/TransY = 0.0 cm
d:Ge/Snout/Hole/TransZ = 0.0 cm
d:Ge/Snout/Hole/RotX = 0.0 deg
d:Ge/Snout/Hole/RotY = 0.0 deg
d:Ge/Snout/Hole/RotZ = 0.0 deg
d:Ge/Snout/Hole/SPhi = 0.0 deg
d:Ge/Snout/Hole/DPhi = 360.0 deg

# Snout: Brass Cone

s:Ge/Snout/BrassCone/Parent = "Snout"
s:Ge/Snout/BrassCone/Type = "G4Cons"
s:Ge/Snout/BrassCone/Material = "Brass"
d:Ge/Snout/BrassCone/RMin1 = 3.5 cm
d:Ge/Snout/BrassCone/RMax1 = 6.0 cm
```
d: Ge/BrassCone/RMin2 = 3.8 cm
d: Ge/BrassCone/RMax2 = 6.0 cm
d: Ge/BrassCone/HL = 3.5 cm
d: Ge/BrassCone/TransX = 0.0 cm
d: Ge/BrassCone/TransY = 0.0 cm
d: Ge/BrassCone/TransZ = -7.0 cm
d: Ge/BrassCone/RotX = 0.0 deg
d: Ge/BrassCone/RotY = 0.0 deg
d: Ge/BrassCone/RotZ = 0.0 deg
d: Ge/BrassCone/SPhi = 0.0 deg
d: Ge/BrassCone/DPhi = 360.0 deg

# Aperture

a: Ge/Aperture/Type = "TsAperture"
a: Ge/Aperture/Parent = "Snout"
a: Ge/Aperture/InputFile = "ApertureFileIn.ap"
a: Ge/Aperture/FileFormat = "XYCoordinates"
b: Ge/Aperture/PrintPoints = "True"
a: Ge/Aperture/Material = "Brass"
d: Ge/Aperture/RMax = 4.5 cm
d: Ge/Aperture/HL = 2.5 cm
d: Ge/Aperture/TransX = 0.0 cm
d: Ge/Aperture/TransY = 0.0 cm
d: Ge/Aperture/TransZ = -13.0 cm
d: Ge/Aperture/RotX = 0.0 deg
d: Ge/Aperture/RotY = 0.0 deg
d: Ge/Aperture/RotZ = 0.0 deg
d: Ge/Aperture/LowerEdge = Ge/Aperture/TransZ - Ge/Aperture/HL cm
s: Ge/Aperture/Message = "Constructing Aperture"

# Compensator

a: Ge/Compensator/Type = "TsCompensator"
a: Ge/Compensator/Parent = "Snout"
a: Ge/Compensator/Material = "CompensatorLucite"
d: Ge/Compensator/RMax = 5.5 cm
d: Ge/Compensator/TransX = 0.0 cm
d: Ge/Compensator/TransY = 0.0 cm
d: Ge/Compensator/Thickness = 0.0 cm # will be reset to actual thickness when compensator is read in
d: Ge/Compensator/HL = 0.5 * Ge/Compensator/Thickness cm
d: Ge/Compensator/TransZ = Ge/Aperture/LowerEdge - Ge/Compensator/HL cm
d: Ge/Compensator/LowerEdge = Ge/Compensator/TransZ - Ge/Compensator/HL cm
d: Ge/Compensator/RotX = 0.0 deg
d: Ge/Compensator/RotY = 0.0 deg
d: Ge/Compensator/RotZ = 0.0 deg
a: Ge/Compensator/InputFile = "CompensatorFileInRowsDepths.rc"
a: Ge/Compensator/FileFormat = "RowsAndDepths"
a: Ge/Compensator/Method = "ExtrudedSolid" # Polyhedra, ExtrudedSolid, _
__SubtractionCylinders or UnionCylinders
d: Ge/Compensator/XTolerance = 1.0 mm
d: Ge/Compensator/YTolerance = 1.0 mm
b: Ge/Compensator/PrintPoints = "True"
s: Ge/Compensator/Message = "Constructing Compensator"
ScatteringNozzle_run.txt

# Use scattering nozzle to generate spread out Bragg peak

includeFile = ScatteringNozzle.txt TsAnalysisWaterPhantom.txt

Ge/RMW_1/RotZ = Tf/RMW_Rotation/Value deg
So/Example/NumberOfHistoriesInRun = 3
Ge/CheckForOverlaps = "t"
Tf/TimelineEnd = 100.0 ms
So/Example/NumberOfSequentialTimes = 100
Tf/RMW_Rotation/Function = "Linear deg"
Tf/RMW_Rotation/Rate = 3.6 deg/ms
Tf/RMW_Rotation/StartValue = 0.0 deg
Tf/RMW_Rotation/RepetitionInterval = 100. ms
b:Ts/PauseBeforeQuit = "False"
i:Ts/ShowHistoryCountAtInterval = 0

# Setting first scatterers:
s:Tf/Scatterer1/L1/Function = "Step"
dv:Tf/Scatterer1/L1/Times = 5 20 40 60 80 100 ms
dv:Tf/Scatterer1/L1/Values = 5 Ge/Scatterer1/RotZ_OutOfBeam Ge/Scatterer1/RotZ_InBeam Ge/Scatterer1/RotZ_OutOfBeam Ge/Scatterer1/RotZ_InBeam Ge/Scatterer1/RotZ_OutOfBeam deg

g:Tf/Scatterer1/L2/Function = "Step"
dv:Tf/Scatterer1/L2/Times = 2 30 100 ms
dv:Tf/Scatterer1/L2/Values = 2 Ge/Scatterer1/RotZ_InBeam Ge/Scatterer1/RotZ_OutOfBeam deg

g:Tf/Scatterer1/L3/Function = "Step"
dv:Tf/Scatterer1/L3/Times = 4 20 40 80 100 ms
dv:Tf/Scatterer1/L3/Values = 4 Ge/Scatterer1/RotZ_InBeam Ge/Scatterer1/RotZ_OutOfBeam Ge/Scatterer1/RotZ_OutOfBeam deg

Ge/Scatterer1/Lollipop1 = Tf/Scatterer1/L1/Value deg
Ge/Scatterer1/Lollipop2 = Tf/Scatterer1/L2/Value deg
Ge/Scatterer1/Lollipop3 = Tf/Scatterer1/L3/Value deg

# Setting second scatterer:
s:Tf/Scatterer2/Function = "Step"
dv:Tf/Scatterer2/Times = 2 50 100 ms
dv:Tf/Scatterer2/Values = 2 Ge/Scatterer2/RotZForS2 Ge/Scatterer2/RotZForS1 deg

d:Ge/Scatterer2/Holder/RotZ = Tf/Scatterer2/Value deg

# Show time feature steps on console
i:Tf/Verbosity = 1
CHAPTER 25

SpecialComponents

CAD.txt

# Builds components on-the-fly from information in CAD files.
# The first component is built from the PLY format.
# The second component is built from the STL format.

d:Ge/World/HLX = 10 cm
d:Ge/World/HLY = 10 cm
d:Ge/World/HLZ = 10 cm
d:Ge/BeamPosition/TransZ = -7 cm
d:Ge/BeamPosition/RotX = 0 deg

d:Ge/CADVolumePly/Parent = "World"
d:Ge/CADVolumePly/Material = "Brass"
d:Ge/CADVolumePly/Type = "TsCAD"
s:Ge/CADVolumePly/InputFile = "Aperture_cm"
s:Ge/CADVolumePly/FileFormat = "ply"
d:Ge/CADVolumePly/Units = 0.5 cm
d:Ge/CADVolumePly/TransX = -4 cm
d:Ge/CADVolumePly/TransY = -4 cm
d:Ge/CADVolumePly/TransZ = -6.5 cm
d:Ge/CADVolumePly/RotX = 0 deg
d:Ge/CADVolumePly/RotY = 0 deg
d:Ge/CADVolumePly/RotZ = 0 deg

d:Ge/CADVolumeStl/Parent = "World"
s:Ge/CADVolumeStl/Material = "Lead"
s:Ge/CADVolumeStl/Type = "TsCAD"
s:Ge/CADVolumeStl/InputFile = "Aperture_cm"
s:Ge/CADVolumeStl/FileFormat = "stl"
d:Ge/CADVolumeStl/Units = 0.5 cm
d:Ge/CADVolumeStl/TransX = -4 cm
DipoleMagnet.txt

# Dipole magnet

```plaintext
d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "True"
s:Ge/Nozzle/Type = "Group"
s:Ge/Nozzle/Parent = "World"
s:Ge/Dipole/Type = "TsBox"
s:Ge/Dipole/Parent = "Nozzle"
s:Ge/Dipole/Material = "G4_AIR"
s:Ge/Dipole/HLX = 10 cm
s:Ge/Dipole/HLY = 10 cm
ds:Ge/Dipole/HLZ = 20 cm
s:Ge/Dipole/Field = "DipoleMagnet"
```

```plaintext
u:Ge/Dipole/MagneticFieldDirectionX = 0.0
u:Ge/Dipole/MagneticFieldDirectionY = 1.0
```
MultiLeafCollimator.txt

```plaintext
# Multileaf Collimator

d:Ge/World/HLX  = 2. m
d:Ge/World/HLY  = 2. m
d:Ge/World/HLZ  = 2. m
s:Ge/World/Material = "G4_Galactic"
b:Ge/World/Invisible = "TRUE"

# Multi-leaf Collimator

s:Ge/MultiLeafCollimatorA/Type  = "TsMultiLeafCollimator"
s:Ge/MultiLeafCollimatorA/Parent = "World"
```

MultiLeafCollimator.txt
s:Ge/MultiLeafCollimatorA/Material = "Aluminum"
d:Ge/MultiLeafCollimatorA/TransX = 0.0 cm
d:Ge/MultiLeafCollimatorA/TransY = 0.0 cm
d:Ge/MultiLeafCollimatorA/TransZ = 0.0 cm
d:Ge/MultiLeafCollimatorA/RotX = 0.0 deg
d:Ge/MultiLeafCollimatorA/RotY = 0.0 deg
d:Ge/MultiLeafCollimatorA/RotZ = 0.0 deg
s:Ge/MultiLeafCollimatorA/DrawingStyle = "Solid"
b:Ge/MultiLeafCollimatorA/PrintInformation = "True"
d:Ge/MultiLeafCollimatorA/MaximumLeafOpen = 5.0 cm
d:Ge/MultiLeafCollimatorA/Thickness = 5.0 cm
d:Ge/MultiLeafCollimatorA/Length = 6.0 cm
dv:Ge/MultiLeafCollimatorA/Widths = 5 1.5 0.5 0.5 0.5 1.5 cm
dv:Ge/MultiLeafCollimatorA/XMinusLeavesOpen = 5 0.0 -0.3 -0.2 -0.5 0.0 cm
dv:Ge/MultiLeafCollimatorA/XPlusLeavesOpen = 5 0.0 0.3 0.2 0.5 0.0 cm

# Physics and Beam Section

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 230.0 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 1.0 cm
d:So/Example/BeamPositionSpreadY = 1.0 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 5
sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
b:Ts/PauseBeforeQuit = "True"
s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 600
i:Gr/ViewA/WindowSizeY = 600
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
g:Gr/ViewA/Projection = "Orthogonal"
d:Gr/ViewA/PerspectiveAngle = 20 deg
u:Gr/ViewA/Zoom = 1.6
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
b:Gr/ViewA/IncludeAxes = "true"
s:Gr/ViewA/AxesComponent = "MultiLeafCollimatorA"
d:Gr/ViewA/AxesSize = 0.1 m
# Multileaf collimator moving in a time sequence

```
includeFile = MultiLeafCollimator.txt

dv:Ge/MultiLeafCollimatorA/XPlusLeavesOpen = 5 0.0 Tp/LeafXPlus1/Value Tp/LeafXPlus2/Value 0.0 cm
dv:Ge/MultiLeafCollimatorA/XMinusLeavesOpen = 5 0.0 Tp/LeafXMinus1/Value Tp/LeafXMinus2/Value 0.0 cm

s:Tp/LeafXPlus0/Function = "Linear cm"
d:Tp/LeafXPlus0/Rate = 0.02 cm/ms
d:Tp/LeafXPlus0/StartValue = 0.0 cm
d:Tp/LeafXPlus0/RepetitionInterval = 100.0 ms

s:Tp/LeafXMinus0/Function = "Linear cm"
d:Tp/LeafXMinus0/Rate = 0.01 cm/ms
d:Tp/LeafXMinus0/StartValue = -0.5 cm
d:Tp/LeafXMinus0/RepetitionInterval = 100.0 ms

s:Tp/LeafXPlus1/Function = "Linear cm"
d:Tp/LeafXPlus1/Rate = 0.04 cm/ms
d:Tp/LeafXPlus1/StartValue = 0.5 cm
d:Tp/LeafXPlus1/RepetitionInterval = 100.0 ms

s:Tp/LeafXMinus1/Function = "Linear cm"
d:Tp/LeafXMinus1/Rate = 0.01 cm/ms
d:Tp/LeafXMinus1/StartValue = 0.0 cm
d:Tp/LeafXMinus1/RepetitionInterval = 100.0 ms

s:Tp/LeafXPlus2/Function = "Linear cm"
d:Tp/LeafXPlus2/Rate = -0.02 cm/ms
d:Tp/LeafXPlus2/StartValue = 0.5 cm
d:Tp/LeafXPlus2/RepetitionInterval = 100.0 ms

s:Tp/LeafXMinus2/Function = "Linear cm"
d:Tp/LeafXMinus2/Rate = -0.03 cm/ms
d:Tp/LeafXMinus2/StartValue = 0.0 cm
d:Tp/LeafXMinus2/RepetitionInterval = 100.0 ms

d:Tp/TimelineEnd = 100.0 ms
i:Tp/NumberOfSequentialTimes = 60
i:Tp/Verbosity = 1
b:Ts/PauseBeforeQuit = "False"

i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
u:Gr/ViewA/Zoom = 2.
d:Gr/ViewA/Theta = Tp/Rot1/Value deg
d:Gr/ViewA/Phi = 90 deg + Tp/Rot1/Value
b:Gr/ViewA/CopyOpenGLToEPS = "False"

s:Tp/Rotate/Function = "Linear deg"
d:Tp/Rotate/Rate = 1. deg/ms
d:Tp/Rotate/StartValue = -40.0 deg
d:Tp/Rotate/RepetitionInterval = 100. ms
```
```
: Tf/Rot1/Function = "Step"
: Tf/Rot1/Times = 2 40 100 ms
: Tf/Rot1/Values = 2 0. Tf/Rotate/Value deg
: Tf/Rot1/RepetitionInterval = 100. ms

MultiWire_Chamber.txt

* Multiwire chamber

: Ge/World/HLX = 50. cm
: Ge/World/HLY = 60. cm
: Ge/World/HLZ = 20. cm

: Ph/Default/Modules = 1 "g4em-standard_opt0"
: Ts/PauseBeforeQuit = "True"

: Ge/WireChamberA/Parent = "World"
: Ge/WireChamberA/Type = "TsMultiWireChamber"
: Ge/WireChamberA/HLX = 30.0 cm
: Ge/WireChamberA/HLY = 30.0 cm
: Ge/WireChamberA/HLZ = 10.0 cm
: Ge/WireChamberA/Material = "Air"
: Ge/WireChamberA/TransX = 0.0 cm
: Ge/WireChamberA/TransY = 0.0 cm
: Ge/WireChamberA/TransZ = 0.0 cm
: Ge/WireChamberA/RotX = 0.0 deg
: Ge/WireChamberA/RotY = 0.0 deg
: Ge/WireChamberA/RotZ = 0.0 deg
: Ge/WireChamberA/PrintInformation = "True"
: Ge/WireChamberA/NbOfLayers = 2

: Ge/WireChamberA/Layer1/RMin = 0.0 cm
: Ge/WireChamberA/Layer1/RMax = 1.0 cm
: Ge/WireChamberA/Layer1/Material = "Brass"
: Ge/WireChamberA/Layer1/HL = 20.0 cm
: Ge/WireChamberA/Layer1/Align = "X"
: Ge/WireChamberA/Layer1/Displacement = 3 -10 0 10 cm
: Ge/WireChamberA/Layer1/PosZ = 5.0 cm
: Ge/WireChamberA/Layer1/DrawingStyle = "FullWireFrame"
: Ge/WireChamberA/Layer1/Color = "red"

: Ge/WireChamberA/Layer2/RMin = 0.5 cm
: Ge/WireChamberA/Layer2/RMax = 1.0 cm
: Ge/WireChamberA/Layer2/Material = "Lexan"
: Ge/WireChamberA/Layer2/HL = 20.0 cm
: Ge/WireChamberA/Layer2/Align = "Y"
: Ge/WireChamberA/Layer2/Displacement = 5 -20 -10 0 10 20 cm
: Ge/WireChamberA/Layer2/PosZ = -5.0 cm
: Ge/WireChamberA/Layer2/DrawingStyle = "Solid"

: Gr/ViewA/Type = "OpenGL"
: Gr/ViewA/WindowSizeX = 600
: Gr/ViewA/WindowSizeY = 600
: Gr/ViewA/Theta = 55 deg
: Gr/ViewA/Phi = 20 deg
```
TOPAS Documentation, Release 3.0

Propeller.txt

# Propeller

d:Ge/World/HLX = 30. cm
d:Ge/World/HLY = 30. cm
d:Ge/World/HLZ = 10. cm

s:Ge/PropellerA/Type = "TsPropeller"
s:Ge/PropellerA/Parent = "World"
d:Ge/PropellerA/TransX = 0.0 cm
d:Ge/PropellerA/TransY = 0.0 cm
d:Ge/PropellerA/TransZ = 0.0 cm
d:Ge/PropellerA/RotX = 0.0 deg
d:Ge/PropellerA/RotY = 0.0 deg
d:Ge/PropellerA/RotZ = 0.0 deg
b:Ge/PropellerA/Invisible = "TRUE"
b:Ge/PropellerA/PrintInformation = "True"
i:Ge/PropellerA/NbOfBlades = 4
d:Ge/PropellerA/Rin = 10.0 mm
d:Ge/PropellerA/Rout = 127.5 mm

s:Ge/PropellerA/Blade/Material = "Parent" #Set to Parent
s:Ge/PropellerA/Blade/Color = "skyblue"
s:Ge/PropellerA/Blade/DrawingStyle = "Solid"
i:Ge/PropellerA/Blade/VisSegsPerCircle = 360
d:Ge/PropellerA/Blade/MaxStepSize = 10 mm

dv:Ge/PropellerA/Thickness = 10
0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 mm
dv:Ge/PropellerA/Angles = 10
63.15 54.15 46.45 40.65 35.85 31.8 28.1 24.725 21.8 19.1 deg

sv:Ge/PropellerA/Materials = 10
"Lexan" "G4_WATER" "G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE"
"G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE" "G4_POLYVINYL_ACETATE"

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
s:So/Example/Type = "Beam"
**TOPAS Documentation, Release 3.0**

```
[s:So/Example/Component] = "BeamPosition"
[s:So/Example/BeamParticle] = "proton"
[d:So/Example/BeamEnergy] = 169.23 MeV
[u:So/Example/BeamEnergySpread] = 0.757504
[s:So/Example/BeamPositionDistribution] = "Gaussian"
[s:So/Example/BeamPositionCutoffShape] = "Ellipse"
[d:So/Example/BeamPositionCutoffX] = 10. cm
[d:So/Example/BeamPositionCutoffY] = 10. cm
[d:So/Example/BeamPositionSpreadX] = 0.65 cm
[d:So/Example/BeamPositionSpreadY] = 0.65 cm
[s:So/Example/BeamAngularDistribution] = "Gaussian"
[d:So/Example/BeamAngularCutoffX] = 90. deg
[d:So/Example/BeamAngularCutoffY] = 90. deg
[d:So/Example/BeamAngularSpreadX] = 0.0032 rad
[d:So/Example/BeamAngularSpreadY] = 0.0032 rad
[i:So/Example/NumberOfHistoriesInRun] = 2

b:Ts/PauseBeforeQuit = "True"

[a:Gr/ViewA/Type] = "OpenGL"
[i:Gr/ViewA/WindowSizeX] = 600
[i:Gr/ViewA/WindowSizeY] = 600
[d:Gr/ViewA/Theta] = 55 deg
[d:Gr/ViewA/Phi] = 20 deg
[a:Gr/ViewA/Projection] = "Perspective"
[d:Gr/ViewA/PerspectiveAngle] = 20 deg
[u:Gr/ViewA/Zoom] = 2.
[b:Gr/ViewA/HiddenLineRemovalForTrajectories] = "True"
[b:Gr/ViewA/IncludeAxes] = "true"
[a:Gr/ViewA/AxesComponent] = "PropellerA"
[d:Gr/ViewA/AxesSize] = 0.1 m

# Propeller with continuous rotation
#includeFile = Propeller.txt
$b:Ts/PauseBeforeQuit = "False"

d:Ge/PropellerA/RotZ = Tf/PropellerRot/Value deg

[s:Tf/PropellerRot/Function] = "Linear deg"
[d:Tf/PropellerRot/Rate] = 3.6 deg/ms
[d:Tf/PropellerRot/StartValue] = 0.0 deg
[d:Tf/PropellerRot/RepetitionInterval] = 100.0 ms

[d:Tf/TimelineEnd] = 100.0 ms
[l:Tf/NumberOfSequentialTimes] = 100
[l:Tf/Verbosity] = 1
```

**Propeller _ContinuousRotation_.txt**

```
#includeFile = Propeller.txt

$b:Ts/PauseBeforeQuit = "False"

d:Ge/PropellerA/RotZ = Tf/PropellerRot/Value deg

[s:Tf/PropellerRot/Function] = "Linear deg"
[d:Tf/PropellerRot/Rate] = 3.6 deg/ms
[d:Tf/PropellerRot/StartValue] = 0.0 deg
[d:Tf/PropellerRot/RepetitionInterval] = 100.0 ms

[d:Tf/TimelineEnd] = 100.0 ms
[l:Tf/NumberOfSequentialTimes] = 100
[l:Tf/Verbosity] = 1
```
Propeller_StepRotation.txt

# Propeller with step rotation

includeFile = Propeller.txt

b:Ts/PauseBeforeQuit = "False"

d:Ge/PropellerA/RotZ = Tf/PropellerRot/Value deg

s:Tf/PropellerRot/Function = "Step"
dv:Tf/PropellerRot/Times = 10 10 20 30 40 50 60 70 80 90 100 ms
dv:Tf/PropellerRot/Values = 10 0 36 72 108 144 180 216 252 288 324 deg

d:Tf/TimelineEnd = 100.0 ms
i:Tf/NumberOfSequentialTimes = 100
i:So/Example/NumberOfHistoriesInRun = 10
i:Tf/Verbosity = 1

PurgingMagnet_move.txt

# A purging magnet, with a non-uniform field,
# defined by the Opera3D format file, PurgMag3D.TABLE
# is moved through a beam, such that the beam encounters
# different areas of the field map at different times.
# The effect is seen as a change over time of beam deflection.

d:Ge/World/HLX = 1.5 m
d:Ge/World/HLY = 1.5 m
d:Ge/World/HLZ = 1.5 m
s:Ge/World/Material = "Vacuum"
b:Ge/World/Invisible = "True"

sv:Ma/Iron/Components = 1"Iron"
v:Ma/Iron/Fractions = 1 1.0
s:Ma/Iron/Density = 7.87 g/cm3
s:Ma/Iron/DefaultColor = "skyblue"

# Nozzle (a group of components)
s:Ge/Nozzle/Type = "Group"
s:Ge/Nozzle/Parent = "World"
d:Ge/Nozzle/TransX = 0. m
d:Ge/Nozzle/TransY = 0. m
d:Ge/Nozzle/TransZ = 0. m
d:Ge/Nozzle/RotX = 0. deg
d:Ge/Nozzle/RotY = 0. deg
d:Ge/Nozzle/RotZ = 0. deg

d:Ge/BeamPosition/TransZ = 15. cm
d:Ge/BeamPosition/RotX = 180. deg

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "e-"
s:So/Example/BeamEnergy = 50.0 MeV
u:So/Example/BeamEnergySpread = 0.0
s:So/Example/BeamPositionDistribution = "None"
s:So/Example/BeamAngularDistribution = "None"

s:Ge/MagGroup/Type = "Group"
s:Ge/MagGroup/Parent = "Nozzle"
d:Ge/MagGroup/TransX = 0.0 m
d:Ge/MagGroup/TransY = 0.0 m
d:Ge/MagGroup/TransZ = 0.0 m

# Topas will crash, if the following is used
# d:Ge/MagGroup/TransZ = Tf/BackStep/Value mm

d:Ge/MagGroup/RotX = 0.0 deg
d:Ge/MagGroup/RotY = 0.0 deg
d:Ge/MagGroup/RotZ = 0.0 deg

s:Ge/V3DBox/Field3D/Type = "TsBox"
s:Ge/V3DBox/Field3D/Field = "MappedMagnet"
s:Ge/V3DBox/Field3D/MagneticField3DTable = "PurgMag3D.TABLE"

# --- done reading
# --- assumed the order: x, y, z, Bx, By, Bz
# --- Min values x,y,z: -5 -5 -26 cm
# --- Max values x,y,z: 5 17 10 cm
#
# After reordering if necessary
# --- Min values x,y,z: -5 -5 -26 cm
# --- Max values x,y,z: 5 17 10 cm
# --- Diff values x,y,z (range): 10 22 36 cm in z

s:Ge/V3DBox/Field3D/Parent = "MagGroup"
s:Ge/V3DBox/Field3D/Material = "Vacuum"
d:Ge/V3DBox/Field3D/HLX = 5 cm
d:Ge/V3DBox/Field3D/HLY = 17 cm
d:Ge/V3DBox/Field3D/HLZ = 26 cm
d:Ge/V3DBox/Field3D/TransX = 0.0 mm
d:Ge/V3DBox/Field3D/TransY = 0.0 mm
d:Ge/V3DBox/Field3D/TransZ = Tf/BackForward/Value mm
#d:Ge/V3DBox/Field3D/TransZ = Tf/BackStep/Value mm
#d:Ge/V3DBox/Field3D/TransZ = 4.4 mm
d:Ge/V3DBox/Field3D/RotX = 0.0 deg
d:Ge/V3DBox/Field3D/RotY = 0.0 deg
d:Ge/V3DBox/Field3D/RotZ = 0.0 deg
b:Ge/V3DBox/Field3D/Include = "TRUE"

#d:Ge/Constants/SSD = -50.0 cm

s:Ge/MeasureVolume/Type = "TsBox"
s:Ge/MeasureVolume/Parent = "World"
s:Ge/MeasureVolume/Material = "World"
d:Ge/MeasureVolume/HLX = 140.0 cm
d:Ge/MeasureVolume/HLY = 140.0 cm
d:Ge/MeasureVolume/HLZ = 0.5 cm
d:Ge/MeasureVolume/TransX = 0.0 cm
d:Ge/MeasureVolume/TransY = 0.0 cm

# -1 * (SSD + HLZ)
d:Ge/MeasureVolume/TransZ = -50.5 cm
d:Ge/MeasureVolume/RotX = 0.0 deg
d:Ge/MeasureVolume/RotY = 0.0 deg
d:Ge/MeasureVolume/RotZ = 0.0 deg
s:Ge/Gap1/Type = "G4Trd"
s:Ge/Gap1/Parent = "Nozzle"
#s:Ge/Gap1/Parent = V3DBox
s:Ge/Gap1/Material = "Iron"
d:Ge/Gap1/HLX1 = 5.0 cm
d:Ge/Gap1/HLX2 = 9.185 cm
d:Ge/Gap1/HLY1 = 5.0 cm
d:Ge/Gap1/HLY2 = 5.0 cm
d:Ge/Gap1/HLZ = 5.725 cm
d:Ge/Gap1/TransX = -9.55 cm
d:Ge/Gap1/TransY = 0.0 cm
d:Ge/Gap1/TransZ = -6.89 cm
d:Ge/Gap1/RotX = 0.0 deg
d:Ge/Gap1/RotY = 0.0 deg
d:Ge/Gap1/RotZ = 0.0 deg
s:Ge/Gap1/DrawingStyle = "FullWireFrame"
b:Ge/Gap1/Include = "FALSE"

s:Ge/Gap2/Type = "G4Trd"
s:Ge/Gap2/Parent = "Nozzle"
#s:Ge/Gap2/Parent = "V3DBox"
s:Ge/Gap2/Material = "Iron"
d:Ge/Gap2/HLX1 = 5.0 cm
d:Ge/Gap2/HLX2 = 9.185 cm
d:Ge/Gap2/HLY1 = 5.0 cm
d:Ge/Gap2/HLY2 = 5.0 cm
d:Ge/Gap2/HLZ = 5.725 cm
d:Ge/Gap2/TransX = 9.55 cm
d:Ge/Gap2/TransY = 0.0 cm
d:Ge/Gap2/TransZ = -6.89 cm
d:Ge/Gap2/RotX = 0.0 deg
d:Ge/Gap2/RotY = 0.0 deg
d:Ge/Gap2/RotZ = 0.0 deg
b:Ge/Gap2/Include = "FALSE"
s:Ge/Gap2/DrawingStyle = "FullWireFrame"

#1. Available steppers:
s:Ge/V3DBox/Field3D/Stepper = "ExplictEuler"
#ExplicitEuler (default)
#ImplicitEuler
#SimpleRunge
#SimpleHeum
#HelixExplicitEuler
#HelixImplicitEuler
#HelixSimpleRunge
#CashKarpRKF45
#RKG3_Stepper

#2. StepMinimum (1.0 mm by default)
d:Ge/V3DBox/Field3D/StepMinimum = 1.0 mm

#3. DeltaChord (0.1 mm by default)
d:Ge/V3DBox/Field3D/DeltaChord = 0.1 mm

Ph/Default/Modules = 1 "g4em-standard_opt0"
s:Tf/BackStep/Function = "Linear mm"
QuadAndDipoleMagnets.txt

# Beam passes through a quadrupole magnet followed by
# a time-varying dipole magnet.

d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"
s:Ge/World/Material = "Vacuum"

# Nozzle (a group of components)
s:Ge/Nozzle/Type = "Group"
s:Ge/Nozzle/Parent = "World"
s:Ge/Nozzle/TransX = 0. m
d:Ge/Nozzle/TransY = 0. m
d:Ge/Nozzle/TransZ = 0. m
d:Ge/Nozzle/RotX = 0. deg
d:Ge/Nozzle/RotY = 0. deg
d:Ge/Nozzle/RotZ = 0. deg

##################################
# Quadrupole pair for focusing beam spot
##################################

s:Ge/QBox1/Quad/Type = "TsBox"
s:Ge/QBox1/Quad/Parent = "Nozzle"
s:Ge/QBox1/Quad/Material = "G4_AIR"
d:Ge/QBox1/Quad/HLX = 15 cm
d:Ge/QBox1/Quad/HLY = 15 cm
d:Ge/QBox1/Quad/HLZ = 10 cm
d:Ge/QBox1/Quad/TransX = 0.0 cm
d:Ge/QBox1/Quad/TransY = 0.0 cm
d:Ge/QBox1/Quad/TransZ = 150 cm + Ge/QBox1/Quad/HLZ
d:Ge/QBox1/Quad/RotX = 0.0 deg
d:Ge/QBox1/Quad/RotY = 0.0 deg
d:Ge/QBox1/Quad/RotZ = 0.0 deg
s:Ge/QBox1/Quad/Field = "QuadrupoleMagnet"
d:Ge/QBox1/Quad/MagneticFieldGradientX = .1 tesla/cm
d:Ge/QBox1/Quad/MagneticFieldGradientY = .1 tesla/cm

s:Ge/QBox2/Quad/Type = "TsBox"
s:Ge/QBox2/Quad/Parent = "Nozzle"
s:Ge/QBox2/Quad/Material = "G4_AIR"
d:Ge/QBox2/Quad/HLX = 15 cm
d:Ge/QBox2/Quad/HLY = 15 cm
d:Ge/QBox2/Quad/HLZ = 10 cm
d:Ge/QBox2/Quad/TransX = 0.0 cm
d:Ge/QBox2/Quad/TransY = 0.0 cm
d:Ge/QBox2/Quad/TransZ = 140 cm + Ge/QBox2/Quad/HLZ
d:Ge/QBox2/Quad/RotX = 0.0 deg
d:Ge/QBox2/Quad/RotY = 0.0 deg
d:Ge/QBox2/Quad/RotZ = 0.0 deg
s:Ge/QBox2/Quad/Field = "QuadrupoleMagnet"
d:Ge/QBox2/Quad/MagneticFieldGradientX = .1 tesla/cm
d:Ge/QBox2/Quad/MagneticFieldGradientY = .1 tesla/cm

# Dipole for stirring beam direction!
# Dipole for stirring beam direction!

s:Ge/VBox1/Dipole/Type = "TsBox"
s:Ge/VBox1/Dipole/Parent = "Nozzle"
s:Ge/VBox1/Dipole/Material = "G4_AIR"
d:Ge/VBox1/Dipole/HLX = 10 cm
d:Ge/VBox1/Dipole/HLY = 10 cm
d:Ge/VBox1/Dipole/HLZ = 20 cm
d:Ge/VBox1/Dipole/TransX = 0.0 cm
d:Ge/VBox1/Dipole/TransY = 0.0 cm
d:Ge/VBox1/Dipole/TransZ = 50.0 cm + Ge/VBox1/Dipole/HLZ
d:Ge/VBox1/Dipole/RotX = 0.0 deg
d:Ge/VBox1/Dipole/RotY = 0.0 deg
d:Ge/VBox1/Dipole/RotZ = 0.0 deg
s:Ge/VBox1/Dipole/Field = "DipoleMagnet"
w:Ge/VBox1/Dipole/MagneticFieldDirectionX = 0.0
w:Ge/VBox1/Dipole/MagneticFieldDirectionY = 1.0
w:Ge/VBox1/Dipole/MagneticFieldDirectionZ = 0.0
w:Ge/VBox1/Dipole/MagneticFieldStrength = 0.0 tesla

s:Ge/VBox2/Dipole/Type = "TsBox"
s:Ge/VBox2/Dipole/Parent = "Nozzle"
s:Ge/VBox2/Dipole/Material = "G4_AIR"
d:Ge/VBox2/Dipole/HLX = 10 cm
d:Ge/VBox2/Dipole/HLY = 10 cm
d:Ge/VBox2/Dipole/HLZ = 20 cm
d:Ge/VBox2/Dipole/TransX = 0.0 cm
d:Ge/VBox2/Dipole/TransY = 0.0 cm
Ge/VBox2/Dipole/TransZ = 90.0 cm + Ge/VBox2/Dipole/HLZ
Ge/VBox2/Dipole/RotX = 0.0 deg
Ge/VBox2/Dipole/RotY = 0.0 deg
Ge/VBox2/Dipole/RotZ = 0.0 deg
Ge/VBox2/Dipole/Field = "DipoleMagnet"
Ge/VBox2/Dipole/MagneticFieldDirectionX = 1.0
Ge/VBox2/Dipole/MagneticFieldDirectionY = 0.0
Ge/VBox2/Dipole/MagneticFieldDirectionZ = 0.0
Ge/VBox2/Dipole/MagneticFieldStrength = Tf/BField1st/Value tesla

# Target window
# Vacuum Window: Target!
Ge/VacFilm/Type = "TsBox"
Ge/VacFilm/Parent = "World"
Ge/VacFilm/Material = "Lead"
Ge/VacFilm/HLX = 50.0 cm
Ge/VacFilm/HLY = 50.0 cm
Ge/VacFilm/HLZ = 1.0 cm
Ge/VacFilm/TransX = 0. cm
Ge/VacFilm/TransY = 0. cm
Ge/VacFilm/TransZ = -60. cm
Ge/VacFilm/RotX = 0. deg
Ge/VacFilm/RotY = 0.0 deg
Ge/VacFilm/RotZ = 0. deg
Ge/VacFilm/Color = "brown"
Ge/VacFilm/DrawingStyle = "solid"

# Ps source
# Ps source
Ge/BeamPosition/TransZ = Ge/World/HLZ cm
Ge/BeamPosition/RotX = 180. deg
So/Example/Type = "Beam"
So/Example/Component = "BeamPosition"
So/Example/BeamParticle = "proton"
So/Example/BeamEnergy = 40.0 MeV
So/Example/BeamEnergySpread = 0.0
So/Example/BeamPositionDistribution = "Gaussian"
So/Example/BeamPositionCutoffShape = "Ellipse"
Ge/BeamPositionCutoffX = 5.0 cm
Ge/BeamPositionCutoffY = 5.0 cm
Ge/BeamPositionSpreadX = 0.65 cm
Ge/BeamPositionSpreadY = 0.65 cm
So/Example/BeamAngularDistribution = "None"

Tf/BField1st/Function = "Step"
Tf/BField1st/Times = 2 10.0 20.0 ms
Tf/BField1st/Values = 2 Tf/BField1stLeft/Value Tf/BField1stRight/Value tesla

Tf/BField1stLeft/Function = "Linear tesla"
Tf/BField1stLeft/Rate = 0.08 tesla/ms
Tf/BField1stLeft/StartValue = -0.32 tesla
Tf/BField1stLeft/RepetitionInterval = 10.0 ms
Tf/BField1stRight/Function = "Linear tesla"
QuadInMovingNozzle.txt

# Magnet mounted in a rotating nozzle

s:So/Example/Type = "Beam"
s:So/Example/Component = "ExitWindow"
s:So/Example/BeamParticle = "chargedgeantino"
s:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.0
s:So/Example/BeamPositionDistribution = "Flat"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
s:So/Example/BeamPositionCutoffX = 3.0 cm
s:So/Example/BeamPositionCutoffY = 3.0 cm
d:So/Example/BeamAngularDistribution = "Gaussian"
s:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 100

sv:Ph/Default/Modules = 1 "g4em-standard_opt0"

b:Ts/PauseBeforeQuit = "True"

s:Tf/NozzleRotation/Function = "Linear deg"
s:Tf/NozzleRotation/Rate = 15. deg/ms
d:Tf/NozzleRotation/StartValue = .0 deg
d:Tf/NozzleRotation/RepetitionInterval = 360. ms
### QuadrupoleMagnet.txt

```plaintext
# MagnetBox magnet

@d: Ge/World/HLX    = 2.0 m
@d: Ge/World/HLY    = 2.0 m
```
$\text{Ge/World/HLZ} = 2.0 \text{ m}$

$b: \text{Ge/World/Invisible} = \text{"True"}$

$s: \text{Ge/Nozzle/Type} = \text{"Group"}$

$s: \text{Ge/Nozzle/Parent} = \text{"World"}$

$s: \text{Ge/MagnetBox/Type} = \text{"TsBox"}$

$s: \text{Ge/MagnetBox/Parent} = \text{"Nozzle"}$

$s: \text{Ge/MagnetBox/Material} = \text{"G4_AIR"}$

$d: \text{Ge/MagnetBox/HLX} = 10 \text{ cm}$

$d: \text{Ge/MagnetBox/HLY} = 10 \text{ cm}$

$d: \text{Ge/MagnetBox/HLZ} = 20 \text{ cm}$

$d: \text{Ge/MagnetBox/RotY} = 0 \text{ deg}$

$d: \text{Ge/MagnetBox/RotX} = 0 \text{ deg}$

$\text{Ge/MagnetBox/Field} = \text{"DipoleMagnet"}$

$\text{Ge/MagnetBox/MagneticFieldDirectionX} = 1.0$

$\text{Ge/MagnetBox/MagneticFieldDirectionY} = 1.0$

$\text{Ge/MagnetBox/MagneticFieldDirectionZ} = 0.0$

$\text{Ge/MagnetBox/MagneticFieldStrength} = 1.0 \text{ tesla}$

$s: \text{Ge/MagnetBox/Field} = \text{"QuadrupoleMagnet"}$

$d: \text{Ge/MagnetBox/MagneticFieldGradientX} = .1 \text{ tesla/cm}$

$d: \text{Ge/MagnetBox/MagneticFieldGradientY} = .1 \text{ tesla/cm}$

$d: \text{Ge/MagnetBox/TransX} = 0 \text{ cm}$

$d: \text{Ge/BeamPosition/TransZ} = \text{Ge/World/HLZ cm}$

$d: \text{Ge/BeamPosition/RotX} = 180. \text{ deg}$

$s: \text{So/Example/Type} = \text{"Beam"}$

$s: \text{So/Example/Component} = \text{"BeamPosition"}$

$s: \text{So/Example/BeamParticle} = \text{"chargedgeantino"}$

$d: \text{So/Example/BeamEnergy} = 169.23 \text{ MeV}$

$\text{So/Example/BeamEnergySpread} = 0.0$

$s: \text{So/Example/BeamPositionDistribution} = \text{"Flat"}$

$s: \text{So/Example/BeamPositionCutoffShape} = \text{"Ellipse"}$

$d: \text{So/Example/BeamPositionCutoffX} = 3.0 \text{ cm}$

$d: \text{So/Example/BeamPositionCutoffY} = 3.0 \text{ cm}$

$s: \text{So/Example/BeamAngularDistribution} = \text{"Gaussian"}$

$d: \text{So/Example/BeamAngularCutoffX} = 90. \text{ deg}$

$d: \text{So/Example/BeamAngularCutoffY} = 90. \text{ deg}$

$d: \text{So/Example/BeamAngularSpreadX} = 0.0032 \text{ rad}$

$d: \text{So/Example/BeamAngularSpreadY} = 0.0032 \text{ rad}$

$i: \text{So/Example/NumberOfHistoriesInRun} = 100$

$s: \text{Gr/ViewA/Type} = \text{"OpenGL"}$

$i: \text{Gr/ViewA/WindowSizeX} = 1024$

$i: \text{Gr/ViewA/WindowSizeY} = 768$

$d: \text{Gr/ViewA/Theta} = 55 \text{ deg}$

$d: \text{Gr/ViewA/Phi} = 20 \text{ deg}$

$s: \text{Gr/ViewA/Projection} = \text{"Perspective"}$

$d: \text{Gr/ViewA/PerspectiveAngle} = 30 \text{ deg}$

$u: \text{Gr/ViewA/Zoom} = 1.0$

$b: \text{Gr/ViewA/IncludeStepPoints} = \text{"True"}$

$s: \text{Ph/Default/Modules} = 1 \text{ "g4em-standard_opt0"}$

$b: \text{Ts/PauseBeforeQuit} = \text{"True"}$
# Range modulator wheel

b:Ge/World/Invisible = "TRUE"

a:Ge/RangeModulatorA/Type = "TsRangeModulator"
a:Ge/RangeModulatorA/Material = "Parent"
a:Ge/RangeModulatorA/Parent = "World"
d:Ge/RangeModulatorA/TransX = 10.0 cm
d:Ge/RangeModulatorA/TransY = 0.0 cm
d:Ge/RangeModulatorA/TransZ = 0.0 cm
d:Ge/RangeModulatorA/RotX = 0.0 deg
d:Ge/RangeModulatorA/RotY = 0.0 deg
d:Ge/RangeModulatorA/RotZ = 0.0 deg
b:Ge/RangeModulatorA/Invisible = "TRUE"
b:Ge/RangeModulatorA/PrintInformation = "True"

d:Ge/RangeModulatorA/HeightOfUpper = 150 mm
d:Ge/RangeModulatorA/HeightOfMiddle = 1.0 mm
d:Ge/RangeModulatorA/HeightOfLower = 9.0 mm

d:Ge/RangeModulatorA/Shell/Rin = 15.0 cm
d:Ge/RangeModulatorA/Shell/Rout = 15.5 cm
a:Ge/RangeModulatorA/Shell/Material = "Aluminum"
a:Ge/RangeModulatorA/Shell/Color = "grey"
a:Ge/RangeModulatorA/Shell/DrawingStyle = "Solid"
i:Ge/RangeModulatorA/Shell/VisSegsPerCircle = 360

d:Ge/RangeModulatorA/Hub/Rin = 6.0 cm
d:Ge/RangeModulatorA/Hub/Rout = 7.0 cm
a:Ge/RangeModulatorA/Hub/Material = "Aluminum"
a:Ge/RangeModulatorA/Hub/Color = "grey"
a:Ge/RangeModulatorA/Hub/DrawingStyle = "Solid"
i:Ge/RangeModulatorA/Hub/VisSegsPerCircle = 360

# Upper tracks
dv:Ge/RangeModulatorA/Upper/RadialDivisions = 1 11.0 cm
a:Ge/RangeModulatorA/Upper/Track1/Pattern = "LexanBlockT1"
a:Ge/RangeModulatorA/Upper/Track2/Pattern = "NULL"

# Middle tracks
dv:Ge/RangeModulatorA/Middle/RadialDivisions = 1 11.0 cm
a:Ge/RangeModulatorA/Middle/Track1/Pattern = "InterfaceDisk"
a:Ge/RangeModulatorA/Middle/Track2/Pattern = "HoleTrackDisk"

# Lower tracks
dv:Ge/RangeModulatorA/Lower/RadialDivisions = 1 11.0 cm
a:Ge/RangeModulatorA/Lower/Track1/Pattern = "LeadBlockT1"
a:Ge/RangeModulatorA/Lower/Track2/Pattern = "NULL"

# A track pattern: 14 blocks of Lexan
d:Ge/LexanBlockT1/Offset = 0.0 deg
dv:Ge/LexanBlockT1/Angles = 14
  5.00 115.00 146.50 173.2 195.07
  216.15 230.14 243.00 255.5 270.60
  282.20 294.60 306.20 324.00 deg
dv:Ge/LexanBlockT1/Heights = 14
  77.0 82.0 87.0 92.15 95.0
  100.4 106.0 110.2 115.3 119.5
sv:Ge/LexanBlockT1/Materials = 14
"Lexan" "Lexan" "Lexan" "Lexan" "Lexan"
"Lexan" "Lexan" "Lexan" "Lexan" "Lexan"
"Lexan" "Lexan" "Lexan" "Lexan"
#A track pattern of single block
dv:Ge/InterfaceDisk/Angles = 1 0.0 deg
dv:Ge/InterfaceDisk/Heights = 1 1.0 mm
sv:Ge/InterfaceDisk/Materials = 1 "Aluminum"

data:Ge/HoleTrackDisk/Angles = 2 90.0 110.0 deg
data:Ge/HoleTrackDisk/Heights = 2 0.0 1.0 mm
sv:Ge/HoleTrackDisk/Materials = 2 NULL "Aluminum"

#A track pattern of 8 Lead blocks
data:Ge/LeadBlockT1/Angles = 9
  5.00 115.00 146.50 173.2 195.07
  216.15 230.14 243.00 255.5 deg
data:Ge/LeadBlockT1/Heights = 9
  0.890 0.75 0.60 0.52 0.40
  0.30 0.16 0.070 0.0 mm
sv:Ge/LeadBlockT1/Materials = 9
"Lead" "Lead" "Lead" "Lead" "Lead"
"Lead" "Lead" "Lead" "NULL"

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
i:Gr/ViewA/Theta = 30 deg
i:Gr/ViewA/Phi = 20 deg
s:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.5
b:Gr/ViewA/HiddenLineRemovalForTrajectories = "True"
b:Gr/ViewA/IncludeAxes = "true"
s:Gr/ViewA/AxesComponent = "RangeModulatorA"
d:Gr/ViewA/AxesSize = 0.1 m
```
sv:Ph/Default/Modules = 1 "g4em-standard_opt0"
b:Ts/PauseBeforeQuit = "True"
i:Ts/ShowHistoryCountAtInterval = 0

RangeModulator_ConstantBeam.txt

# Range modulator wheel in constant current beam
includeFile = RangeModulator.txt
d:Ge/RangeModulatorA/Rotz = Tf/RMW_Rotation/Value deg
i:So/Example/NumberOfHistoriesInRun = 3
d:Tf/TimelineEnd = 12.0 ms
i:Tf/NumberOfSequentialTimes = 20
s:Tf/RMW_Rotation/Function = "Linear deg"
d:Tf/RMW_Rotation/Rate = 3.6 deg/ms
d:Tf/RMW_Rotation/StartValue = 0.0 deg
d:Tf/RMW_Rotation/RepetitionInterval = 100. ms
b:Ts/PauseBeforeQuit = "False"
i:Ts/ShowHistoryCountAtInterval = 0

# Show time feature steps on console
i:Tf/Verbosity = 1
b:Gr/ViewA/CopyOpenGLToEPS = "False"

RangeModulator_CurrentModulatedBeam.txt

# Range modulator wheel in modulated current beam
includeFile = RangeModulator_ConstantBeam.txt
s:Tf/BeamCurrent/Function = "Step"
dv:Tf/BeamCurrent/Times = 1 10 ms
iv:Tf/BeamCurrent/Values = 1 10
s:Tf/BeamWeight/Function = "Step"
dv:Tf/BeamWeight/Times = 10 1 2 3 4 5 6 7 8 9 10 ms
iv:Tf/BeamWeight/Values = 10 1 4 1 1 2 5 2 3 2 1
i:So/Example/NumberOfHistoriesInRun = Tf/BCM_1/Value
i:Tf/BCM_1/Value = Tf/BeamWeight/Value * Tf/BeamCurrent/Value
```
# Ridge Filter

d:Ge/World/HLX = 0.15 m
d:Ge/World/HLY = 0.15 m
d:Ge/World/HLZ = 0.05 m
b:Ge/World/Invisible = "True"

# Ridge Filter Group

s:Ge/RidgeGroup/Parent = "World"
s:Ge/RidgeGroup/Type = "Group"
d:Ge/RidgeGroup/RotX = 0.0 deg
d:Ge/RidgeGroup/RotY = 0.0 deg
d:Ge/RidgeGroup/RotZ = 0.0 deg
d:Ge/RidgeGroup/TransX = 0.0 cm
d:Ge/RidgeGroup/TransY = 0.0 cm
d:Ge/RidgeGroup/TransZ = 0.0 cm

# Ridge Filter

s:Ge/RidgeFilterA/Type = "TsRidgeFilter"
s:Ge/RidgeFilterA/Parent = "RidgeGroup"
s:Ge/RidgeFilterA/Material = "Aluminum"
d:Ge/RidgeFilterA/TransX = 0.0 cm
d:Ge/RidgeFilterA/TransY = 0.0 cm
d:Ge/RidgeFilterA/TransZ = 0.0 cm
d:Ge/RidgeFilterA/RotX = 0.0 deg
d:Ge/RidgeFilterA/RotY = 0.0 deg
d:Ge/RidgeFilterA/RotZ = 0.0 deg
s:Ge/RidgeFilterA/DrawingStyle = "Solid"
b:Ge/RidgeFilterA/PrintInformation = "True"

# Ridge Geometry: TOPAS will connect points of x and z to build a ridge.
# Then copy a ridge and place it using Displacement parameter

dv:Ge/RidgeFilterA/XPoints = 8
0.0 0.8 1.3 1.8 2.2 2.7 3.2 4.0 mm
dv:Ge/RidgeFilterA/ZPoints = 8
2.4 4.0 9.1 14.0 14.0 9.1 4.0 2.4 mm
d:Ge/RidgeFilterA/Width = 4.0 mm
d:Ge/RidgeFilterA/Length = 1.0 cm
dv:Ge/RidgeFilterA/Displacement = 3 -5.0 0.0 5.0 mm

# Physics, Beam, and Graphic setting

s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "proton"
d:So/Example/BeamEnergy = 230.0 MeV
w:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 10. cm
RotatingMagnet.txt

# A dipole magnet has field strength changing over time
# such that magnet is sweeping back and forth.
# At the same time, the nozzle that contains this magnet is rotating.

d:Ge/World/HLX = 2.0 m
d:Ge/World/HLY = 2.0 m
d:Ge/World/HLZ = 2.0 m
b:Ge/World/Invisible = "TRUE"

# Nozzle (a group of components)
s:Ge/Nozzle/Type = "Group"
s:Ge/Nozzle/Parent = "World"
d:Ge/Nozzle/TransX = 0. m
d:Ge/Nozzle/TransY = 0. m
d:Ge/Nozzle/TransZ = 0. m
d:Ge/Nozzle/RotX = 0. deg
d:Ge/Nozzle/RotY = 0. deg
d:Ge/Nozzle/RotZ = Tf/MagnetRotation/Value deg

s:Tf/MagnetRotation/Function = "Linear deg"
s:Tf/MagnetRotation/Rate = 2. deg/ms
d:Tf/MagnetRotation/StartValue = -0. deg
d:Tf/MagnetRotation/RepetitionInterval = 80.0 ms

##################################################
# Dipole for stirring beam direction!
s:Ge/VBox2/Dipole/Type = "TsBox"
s:Ge/VBox2/Dipole/Parent = "Nozzle"
s:Ge/VBox2/Dipole/Material = "G4_AIR"
d:Ge/VBox2/Dipole/HLX = 10 cm
d:Ge/VBox2/Dipole/HLY = 10 cm
d:Ge/VBox2/Dipole/HLZ = 20 cm
d:Ge/VBox2/Dipole/TransX = 0.0 cm
d:Ge/VBox2/Dipole/TransY = 0.0 cm
d:Ge/VBox2/Dipole/TransZ = 90.0 cm + Ge/VBox2/Dipole/HLZ
d:Ge/VBox2/Dipole/RotX = 0.0 deg
d:Ge/VBox2/Dipole/RotY = 0.0 deg
d:Ge/VBox2/Dipole/RotZ = 0.0 deg
s:Ge/VBox2/Dipole/Field = "DipoleMagnet"
u:Ge/VBox2/Dipole/MagneticFieldDirectionX = 1.0
u:Ge/VBox2/Dipole/MagneticFieldDirectionY = 0.0
u:Ge/VBox2/Dipole/MagneticFieldDirectionZ = 0.0
d:Ge/VBox2/Dipole/MagneticFieldStrength = Tf/BField1st/Value tesla

# Vacuum window for phase space!
# Vacuum Window: Target!
s:Ge/VacFilm/Type = "TsCylinder"
s:Ge/VacFilm/Parent = "World"
s:Ge/VacFilm/Material = "G4_WATER"
d:Ge/VacFilm/RMin = 0.0 cm
d:Ge/VacFilm/RMax = 100.0 cm
d:Ge/VacFilm/HL = 1.0 cm
d:Ge/VacFilm/SPhi = 0.0 deg
d:Ge/VacFilm/DPhi = 360.0 deg
d:Ge/VacFilm/TransX = 0.0 cm
d:Ge/VacFilm/TransY = 0.0 cm
d:Ge/VacFilm/TransZ = 0.0 cm
d:Ge/VacFilm/RotX = 0.0 deg
d:Ge/VacFilm/RotY = 0.0 deg
d:Ge/VacFilm/RotZ = 0.0 deg
s:Ge/VacFilm/Color = "skyblue"
#s:Sc/Plane1/Quantity = "PhaseSpace"
#s:Sc/Plane1/Surface = "VacFilm/ZMinusSurface"
#s:Sc/Plane1/OutputType = "ASCII"
#i:Sc/Plane1/OutputBufferSize = 1000

# Ps source
# Ps source
d:Ge/BeamPosition/TransZ = Ge/World/HLZ cm
d:Ge/BeamPosition/RotX = 180.0 deg
s:Ge/World/Material = "Vacuum"
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "e+"
d:So/Example/BeamEnergy = 40.0 MeV
u:So/Example/BeamEnergySpread = 5.

25.17. RotatingMagnet.txt
So/Example/BeamPositionDistribution = "Gaussian"
So/Example/BeamPositionCutoffShape = "Ellipse"
So/Example/BeamPositionCutoffX = 5.0 cm
So/Example/BeamPositionCutoffY = 5.0 cm
So/Example/BeamPositionSpreadX = 0.65 cm
So/Example/BeamPositionSpreadY = 0.65 cm
So/Example/BeamAngularDistribution = "Gaussian"
So/Example/BeamAngularCutoffX = 90. deg
So/Example/BeamAngularCutoffY = 90. deg
So/Example/BeamAngularSpreadX = 0.005 rad
So/Example/BeamAngularSpreadY = 0.005 rad
So/Example/NumberOfHistoriesInRun = 500

Tf/BField1st/Function = "Step"
Tf/BField1st/Times = 2 10.0 20.0 ms
Tf/BField1st/Values = 2 Tf/BField1stLeft/Value Tf/BField1stRight/Value tesla

Tf/BField1stLeft/Function = "Linear tesla"
Tf/BField1stLeft/Rate = 0.064 tesla/ms
Tf/BField1stLeft/StartValue = -0.32 tesla
Tf/BField1stLeft/RepetitionInterval = 10.0 ms

Tf/BField1stRight/Function = "Linear tesla"
Tf/BField1stRight/Rate = -0.064 tesla/ms
Tf/BField1stRight/StartValue = 0.32 tesla
Tf/BField1stRight/RepetitionInterval = 10.0 ms

Tf/SmallBox/Function = "Linear tesla"
Tf/SmallBox/Rate = 0.1 tesla/ms
Tf/SmallBox/StartValue = 0.0 tesla
Tf/SmallBox/RepetitionInterval = 70. ms

Tf/TimelineEnd = 40.0 ms
Tf/NumberOfSequentialTimes = 10

Gr/ViewA/Type = "OpenGL"
Gr/ViewA/WindowSizeX = 1024
Gr/ViewA/WindowSizeY = 768
Gr/ViewA/Theta = 55 deg
Gr/ViewA/Phi = 20 deg
Gr/ViewA/Projection = "Perspective"
Gr/ViewA/PerspectiveAngle = 30 deg
Gr/ViewA/Zoom = 1.4
Gr/ViewA/TransX = -.2
Gr/ViewA/HiddenLineRemovalForTrajectories = "True"

# Uniform electric field
Ge/World/HLX = 2.0 m
Ge/World/HLY = 2.0 m
Ge/World/HLZ = 2.0 m
Ge/World/Invisible = "True"
Ge/World/Material = "Vacuum"
TOPAS Documentation, Release 3.0

s:Ge/DriftBox/Type = "TsBox"
s:Ge/DriftBox/Parent = "World"
s:Ge/DriftBox/Material = "Vacuum"
d:Ge/DriftBox/HLX = 80 cm
d:Ge/DriftBox/HLY = 80 cm
d:Ge/DriftBox/HLZ = 100 cm
d:Ge/DriftBox/Field = "UniformElectroMagnetic"
u:Ge/DriftBox/ElectricFieldDirectionX = 1.0
u:Ge/DriftBox/ElectricFieldDirectionY = 1.0
u:Ge/DriftBox/ElectricFieldDirectionZ = 0.0
d:Ge/DriftBox/ElectricFieldStrength = 5000 kV/cm
u:Ge/DriftBox/MagneticFieldDirectionX = 0.0
u:Ge/DriftBox/MagneticFieldDirectionY = 1.0
u:Ge/DriftBox/MagneticFieldDirectionZ = 0.0
d:Ge/DriftBox/MagneticFieldStrength = 5.0 tesla
d:Ge/DriftBox/RotZ = Tf/RotationStep/Value deg
d:Ge/BeamPosition/TransZ = Ge/World/HLZ cm
d:Ge/BeamPosition/RotX = 180. deg
i:Tf/Verbosity = 1
d:Tf/TimelineEnd = 30.0 ms
i:Tf/NumberOfSequentialTimes = 40
s:Tf/RotationStep/Function = "Linear deg"
d:Tf/RotationStep/Rate = 5. deg/ms
d:Tf/RotationStep/StartValue = -28.0 deg
d:Tf/RotationStep/RepetitionInterval = 360. ms
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "chargedgeantino"
d:So/Example/BeamEnergy = 169.23 MeV
u:So/Example/BeamEnergySpread = 0.0
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 5.0 cm
d:So/Example/BeamPositionCutoffY = 5.0 cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "None"
i:So/Example/NumberOfHistoriesInRun = 10
s:Gr/ViewA/Type = "OpenGL"
i:Gr/ViewA/WindowSizeX = 1024
i:Gr/ViewA/WindowSizeY = 768
d:Gr/ViewA/Theta = 55 deg
d:Gr/ViewA/Phi = 20 deg
a:Gr/ViewA/Projection = "Perspective"
d:Gr/ViewA/PerspectiveAngle = 30 deg
u:Gr/ViewA/Zoom = 1.
b:Gr/ViewA/IncludeStepPoints = "True"
b:Ts/PauseBeforeQuit = "True"
CHAPTER 26

VarianceReduction

AmbientDoseEquivalent.txt

# Simple shielding example

# Geometry for vrt

d:Ge/World/HLX = 2 m
d:Ge/World/HLY = 2 m
d:Ge/World/HLZ = 2.1 m

d:Ge/ScorerHLX = 1 m
d:Ge/ScorerHLY = 1 m
d:Ge/ScorerHLZ = 1 m

d:Ge/subComponent/Thickness = 20 cm
d:Ge/subComponent/StartPosition = -1.0 * Ge/ScorerHLZ cm

g:Ge/subComponent1/Material = "G4_CONCRETE"
g:Ge/subComponent2/Material = "G4_CONCRETE"
g:Ge/subComponent3/Material = "G4_CONCRETE"
g:Ge/subComponent4/Material = "G4_CONCRETE"
g:Ge/subComponent5/Material = "G4_CONCRETE"
g:Ge/subComponent6/Material = "G4_CONCRETE"
g:Ge/subComponent7/Material = "G4_CONCRETE"
g:Ge/subComponent8/Material = "G4_CONCRETE"
g:Ge/subComponent9/Material = "G4_CONCRETE"
g:Ge/subComponent10/Material = "G4_CONCRETE"
g:Ge/subComponent11/Material = "G4_CONCRETE"

g:Ge/subComponent1/Type = "TsBox"
g:Ge/subComponent1/Parent = "World"
TOPAS Documentation, Release 3.0

326 Chapter 26. VarianceReduction

```c
// Ge/subComponent1/HLX = Ge/ScorerHLX m
// Ge/subComponent1/HLY = Ge/ScorerHLY m
// Ge/subComponent1/HLZ = 0.5 * Ge/subComponent/Thickness cm
// Ge/subComponent1/TransX = 0 cm
// Ge/subComponent1/TransY = 0 cm
// Ge/subComponent1/TransZ = Ge/subComponent/StartPosition + Ge/subComponent1/HLZ cm
// Ge/subComponent1/RotX = 0 deg
// Ge/subComponent1/RotY = 0 deg
// Ge/subComponent1/RotZ = 0 deg

// Ge/subComponent2/Type = "TsBox"
// Ge/subComponent2/Parent = "World"
// Ge/subComponent2/HLX = Ge/ScorerHLX m
// Ge/subComponent2/HLY = Ge/ScorerHLY m
// Ge/subComponent2/HLZ = 0.5 * Ge/subComponent/Thickness cm
// Ge/subComponent2/TransX = 0 cm
// Ge/subComponent2/TransY = 0 cm
// Ge/subComponent2/TransZ = Ge/subComponent2/TransZ + Ge/subComponent1/HLZ cm
// Ge/subComponent2/RotX = 0 deg
// Ge/subComponent2/RotY = 0 deg
// Ge/subComponent2/RotZ = 0 deg

// Ge/subComponent3/Type = "TsBox"
// Ge/subComponent3/Parent = "World"
// Ge/subComponent3/HLX = Ge/ScorerHLX m
// Ge/subComponent3/HLY = Ge/ScorerHLY m
// Ge/subComponent3/HLZ = 0.5 * Ge/subComponent/Thickness cm
// Ge/subComponent3/TransX = 0 cm
// Ge/subComponent3/TransY = 0 cm
// Ge/subComponent3/TransZ = Ge/subComponent3/TransZ + Ge/subComponent2/HLZ cm
// Ge/subComponent3/RotX = 0 deg
// Ge/subComponent3/RotY = 0 deg
// Ge/subComponent3/RotZ = 0 deg

// Ge/subComponent4/Type = "TsBox"
// Ge/subComponent4/Parent = "World"
// Ge/subComponent4/HLX = Ge/ScorerHLX m
// Ge/subComponent4/HLY = Ge/ScorerHLY m
// Ge/subComponent4/HLZ = 0.5 * Ge/subComponent/Thickness cm
// Ge/subComponent4/TransX = 0 cm
// Ge/subComponent4/TransY = 0 cm
// Ge/subComponent4/TransZ = Ge/subComponent4/TransZ + Ge/subComponent3/HLZ cm
// Ge/subComponent4/RotX = 0 deg
// Ge/subComponent4/RotY = 0 deg
// Ge/subComponent4/RotZ = 0 deg

// Ge/subComponent5/Type = "TsBox"
// Ge/subComponent5/Parent = "World"
// Ge/subComponent5/HLX = Ge/ScorerHLX m
// Ge/subComponent5/HLY = Ge/ScorerHLY m
// Ge/subComponent5/HLZ = 0.5 * Ge/subComponent/Thickness cm
// Ge/subComponent5/TransX = 0 cm
// Ge/subComponent5/TransY = 0 cm
// Ge/subComponent5/TransZ = Ge/subComponent5/TransZ + Ge/subComponent4/HLZ cm
// Ge/subComponent5/RotX = 0 deg
// Ge/subComponent5/RotY = 0 deg
// Ge/subComponent5/RotZ = 0 deg
```
d:Ge/subComponent5/RotX = 0 deg
d:Ge/subComponent5/RotY = 0 deg
d:Ge/subComponent5/RotZ = 0 deg

s:Ge/subComponent6/Type = "TsBox"
s:Ge/subComponent6/Parent = "World"
d:Ge/subComponent6/HLX = Ge/ScorerHLX m
d:Ge/subComponent6/HLY = Ge/ScorerHLY m
d:Ge/subComponent6/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent6/TransX = 0 cm
d:Ge/subComponent6/TransY = 0 cm
d:Ge/subComponent6/TransZTmp = Ge/subComponent5/TransZ + Ge/subComponent5/HLZ cm
d:Ge/subComponent6/TransZ = Ge/subComponent6/TransZTmp + Ge/subComponent6/HLZ cm
d:Ge/subComponent6/RotX = 0 deg
d:Ge/subComponent6/RotY = 0 deg
d:Ge/subComponent6/RotZ = 0 deg

s:Ge/subComponent7/Type = "TsBox"
s:Ge/subComponent7/Parent = "World"
d:Ge/subComponent7/HLX = Ge/ScorerHLX m
d:Ge/subComponent7/HLY = Ge/ScorerHLY m
d:Ge/subComponent7/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent7/TransX = 0 cm
d:Ge/subComponent7/TransY = 0 cm
d:Ge/subComponent7/TransZTmp = Ge/subComponent6/TransZ + Ge/subComponent6/HLZ cm
d:Ge/subComponent7/TransZ = Ge/subComponent7/TransZTmp + Ge/subComponent7/HLZ cm
d:Ge/subComponent7/RotX = 0 deg
d:Ge/subComponent7/RotY = 0 deg
d:Ge/subComponent7/RotZ = 0 deg

s:Ge/subComponent8/Type = "TsBox"
s:Ge/subComponent8/Parent = "World"
d:Ge/subComponent8/HLX = Ge/ScorerHLX m
d:Ge/subComponent8/HLY = Ge/ScorerHLY m
d:Ge/subComponent8/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent8/TransX = 0 cm
d:Ge/subComponent8/TransY = 0 cm
d:Ge/subComponent8/TransZTmp = Ge/subComponent7/TransZ + Ge/subComponent7/HLZ cm
d:Ge/subComponent8/TransZ = Ge/subComponent8/TransZTmp + Ge/subComponent8/HLZ cm
d:Ge/subComponent8/RotX = 0 deg
d:Ge/subComponent8/RotY = 0 deg
d:Ge/subComponent8/RotZ = 0 deg

s:Ge/subComponent9/Type = "TsBox"
s:Ge/subComponent9/Parent = "World"
d:Ge/subComponent9/HLX = Ge/ScorerHLX m
d:Ge/subComponent9/HLY = Ge/ScorerHLY m
d:Ge/subComponent9/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent9/TransX = 0 cm
d:Ge/subComponent9/TransY = 0 cm
d:Ge/subComponent9/TransZTmp = Ge/subComponent8/TransZ + Ge/subComponent8/HLZ cm
d:Ge/subComponent9/TransZ = Ge/subComponent9/TransZTmp + Ge/subComponent9/HLZ cm
d:Ge/subComponent9/RotX = 0 deg
d:Ge/subComponent9/RotY = 0 deg
d:Ge/subComponent9/RotZ = 0 deg

s:Ge/subComponent10/Type = "TsBox"
s:Ge/subComponent10/Parent = "World"
# VRT

b:Vr/UseVarianceReduction = "true"
b:Vr/ParticleSplit/Active = "true"
sv:Vr/ParticleSplit/ParticleName = 1 "neutron"
sv:Vr/ParticleSplit/Component = "subComponent1"
sv:Vr/ParticleSplit/SubComponents = 10

"subComponent2"
"subComponent3" "subComponent4"
"subComponent5" "subComponent6"
"subComponent7" "subComponent8"
"subComponent9" "subComponent10" "subComponent11"
s:Vr/ParticleSplit/Type = "ImportanceSampling"

uv:Vr/ParticleSplit/ImportanceValues = 10 2 4 8 16 32 64 128 256 512 512

# Fix binwidth

sv:Sc/scorerFix/OnlyIncludeParticlesNamed = 1 "neutron"
s:Sc/scorerFix/Quantity = "Fluence"
s:Sc/scorerFix/Component = "subComponent11"
s:Sc/scorerFix/OutputFile = "FluenceSpectra_FixedBin"
s:Sc/scorerFix/OutputType = "Binary"
s:Sc/scorerFix/IfOutputFileAlreadyExists = "Overwrite"
l:Sc/scorerFix/EBins = 100
d:Sc/scorerFix/EBinMin = 0.01 MeV
d:Sc/scorerFix/EBinMax = 110.1 MeV

# Logarithmic binwidth

sv:Sc/scorerLog/OnlyIncludeParticlesNamed = 1 "neutron"
s:Sc/scorerLog/Quantity = "Fluence"
s:Sc/scorerLog/Component = "subComponent11"
TOPAS Documentation, Release 3.0

# Score dose to medium

```plaintext
# Score ambient dose equivalent for neutrons
```

```plaintext
# Set the fluence-to-dose conversion factors.
```

```plaintext
# Beam setting
```

```

---

26.1. AmbientDoseEquivalent.txt 329
TOPAS Documentation, Release 3.0

GeometricalParticleSplit.txt

```
# Time features
Tf/Phi/Function = "Linear deg"
Tf/Phi/Rate = 90. deg/ms
Tf/Phi/StartValue = 45 deg
Tf/Phi/RepetitionInterval = 4. ms
Tf/TimelineEnd = 4. ms
Tf/NumberOfSequentialTimes = 4

# Geometry
Ge/World/HLX = 1.0 m
Ge/World/HLY = 1.0 m
Ge/World/HLZ = 1.0 m
Ge/World/Invisible = "TRUE"

# Phantom
Ge/Phantom/Type = "TsSphere"
Ge/Phantom/Parent = "World"
Ge/Phantom/Material = "G4_WATER"
Ge/Phantom/Color = "blue"
Ge/Phantom/RMin = 0.0 cm
Ge/Phantom/RMax = 10.0 cm
Ge/Phantom/SPhi = 0.0 deg
Ge/Phantom/DPhi = 360 deg
Ge/Phantom/STheta = 0 deg
```
\[ a \]: Ge/Phantom/DTheta = 180 deg  
\[ a \]: Ge/Phantom/TransX = 0 cm  
\[ a \]: Ge/Phantom/TransY = 0 cm  
\[ a \]: Ge/Phantom/TransZ = 0 cm  
\[ a \]: Ge/Phantom/RotX = 0 deg  
\[ a \]: Ge/Phantom/RotY = 0 deg  
\[ a \]: Ge/Phantom/RotZ = 0 deg  

# Nozzle

\[ a \]: Ge/Nozzle/Type = "Group"  
\[ a \]: Ge/Nozzle/Parent = "World"  
\[ a \]: Ge/Nozzle/TransX = 0. cm  
\[ a \]: Ge/Nozzle/TransY = 0. cm  
\[ a \]: Ge/Nozzle/TransZ = 0. cm  
\[ a \]: Ge/Nozzle/RotX = Tf/Phi/Value deg  
\[ a \]: Ge/Nozzle/RotY = 0 deg  
\[ a \]: Ge/Nozzle/RotZ = 0 deg  
\[ a \]: Ge/Nozzle/Color = "blue"

# Scatterers

\[ a \]: Ge/Scatterer1/Type = "TsBox"  
\[ a \]: Ge/Scatterer1/Parent = "Nozzle"  
\[ a \]: Ge/Scatterer1/Material = "Lead"  
\[ a \]: Ge/Scatterer1/HLX = 2.0 cm  
\[ a \]: Ge/Scatterer1/HLY = 2.0 cm  
\[ a \]: Ge/Scatterer1/HLZ = 0.25 cm  
\[ a \]: Ge/Scatterer1/TransX = 0. cm  
\[ a \]: Ge/Scatterer1/TransY = 0. cm  
\[ a \]: Ge/Scatterer1/TransZ = -50 cm + Ge/Scatterer1/HLZ  
\[ a \]: Ge/Scatterer1/RotX = 0 deg  
\[ a \]: Ge/Scatterer1/RotY = 0 deg  
\[ a \]: Ge/Scatterer1/RotZ = 0 deg  
\[ a \]: Ge/Scatterer1/Color = "red"

\[ a \]: Ge/Scatterer2/Type = "TsBox"  
\[ a \]: Ge/Scatterer2/Parent = "Nozzle"  
\[ a \]: Ge/Scatterer2/Material = "Lead"  
\[ a \]: Ge/Scatterer2/HLX = 4.0 cm  
\[ a \]: Ge/Scatterer2/HLY = 4.0 cm  
\[ a \]: Ge/Scatterer2/HLZ = 0.25 cm  
\[ a \]: Ge/Scatterer2/TransX = 0. cm  
\[ a \]: Ge/Scatterer2/TransY = 0. cm  
\[ a \]: Ge/Scatterer2/TransZ = -40 cm + Ge/Scatterer2/HLZ  
\[ a \]: Ge/Scatterer2/RotX = 0 deg  
\[ a \]: Ge/Scatterer2/RotY = 0 deg  
\[ a \]: Ge/Scatterer2/RotZ = 0 deg  
\[ a \]: Ge/Scatterer2/Color = "red"

# Geometry for vrt

\[ a \]: Ge/VrtParallelWorld/Type = "TsBox"  
\[ a \]: Ge/VrtParallelWorld/Parent = "Nozzle"
Variance reduction

# Beam settings

s:So/Example/Type = "Beam"
So/Example/Component = "BeamPosition"
So/Example/BeamParticle = "proton"
So/Example/BeamEnergy = 170 MeV
So/Example/BeamEnergySpread = 0.757504
So/Example/BeamPositionDistribution = "Gaussian"
So/Example/BeamPositionCutoffShape = "Ellipse"
So/Example/BeamPositionCutoffX = 0.15 cm
So/Example/BeamPositionCutoffY = 0.15 cm
So/Example/BeamPositionSpreadX = 0.65 cm
So/Example/BeamPositionSpreadY = 0.65 cm
So/Example/BeamAngularDistribution = "Gaussian"
So/Example/BeamAngularCutoffX = 90. deg
So/Example/BeamAngularCutoffY = 90. deg
So/Example/BeamAngularSpreadX = 0.0032 rad
So/Example/BeamAngularSpreadY = 0.0032 rad

Ge/BeamPosition/Parent = "Nozzle"
Ge/BeamPosition/RotX = 0 deg
Ge/BeamPosition/TransZ = -60. cm

##########################
# Scorer
##########################
Sc/scorer/Quantity = "DoseToMedium"
Sc/scorer/Component = "Phantom"
Sc/scorer/Report = 3 "Sum" "Mean" "Variance"
Sc/scorer/OutputType = "CSV"
Sc/scorer/IfOutputFileAlreadyExists = "Overwrite"

##########################
# Histories
##########################
So/Example/NumberOfHistoriesInRun = 100
Ts/ShowHistoryCountAtInterval = 100
Ts/PauseBeforeQuit = "False"

##########################
# Visualization
##########################
Gr/viewer/Type = "OpenGl"
Gr/viewer/Zoom = 0.5
Gr/viewer/WindowSizeX = 900
Gr/viewer/WindowSizeY = 900
Gr/viewer/WindowPosX = 0
Gr/viewer/WindowPosY = 0
Gr/viewer/Projection = "Orthogonal"
Gr/viewer/Theta = -90. deg
Gr/viewer/Phi = 0. deg
Gr/viewer/IncludeAxes = "true"
Gr/viewer/AxesComponent = "World"
Gr/OnlyIncludeParticlesNamed = 1 "proton"
Gr/viewer/AxesSize = 20 cm
# Simple shielding example

d:Ge/World/HLX = 2 m
d:Ge/World/HLY = 2 m
d:Ge/World/HLZ = 2.1 m

s:Ge/wall/Type = "TsBox"
s:Ge/wall/Parent = "World"
s:Ge/wall/Material = "G4_CONCRETE"
d:Ge/wall/HLX = 1 m
d:Ge/wall/HLY = 1 m
d:Ge/wall/HLZ = 1 m
d:Ge/wall/TransX = 0 m
d:Ge/wall/TransY = 0 m
d:Ge/wall/TransZ = 0 m
d:Ge/wall/RotX = 0 deg
d:Ge/wall/RotY = 0 deg
d:Ge/wall/RotZ = 0 deg
s:Ge/wall/Color = "red"
i:Ge/wall/XBins = 1
i:Ge/wall/YBins = 1
i:Ge/wall/ZBins = 10

# Geometry for vrt

d:Ge/subComponent/Thickness = 20 cm
d:Ge/subComponent/StartPosition = -1.0 * Ge/wall/HLZ cm

s:Ge/VrtParallelWorld/Type = "TsBox"
s:Ge/VrtParallelWorld/Parent = "World"
d:Ge/VrtParallelWorld/HLX = 1.1 m
d:Ge/VrtParallelWorld/HLY = 1.1 m
d:Ge/VrtParallelWorld/HLZ = 1.1 m
d:Ge/VrtParallelWorld/TransX = 0 cm
d:Ge/VrtParallelWorld/TransY = 0 cm
d:Ge/VrtParallelWorld/TransZ = 0 cm
d:Ge/VrtParallelWorld/RotX = 0 deg
d:Ge/VrtParallelWorld/RotY = 0 deg
d:Ge/VrtParallelWorld/RotZ = 0 deg
b:Ge/VrtParallelWorld/IsParallel = "true"

s:Ge/subComponent1/Type = "TsBox"
s:Ge/subComponent1/Parent = "VrtParallelWorld"
d:Ge/subComponent1/HLX = Ge/Wall/HLX m
d:Ge/subComponent1/HLY = Ge/Wall/HLY m
d:Ge/subComponent1/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent1/TransX = 0 cm
d:Ge/subComponent1/TransY = 0 cm
d:Ge/subComponent1/TransZ = Ge/subComponent/StartPosition + Ge/subComponent1/HLZ...
cm
d:Ge/subComponent1/RotX = 0 deg
d:Ge/subComponent1/RotY = 0 deg
d:Ge/subComponent1/RotZ = 0 deg
b:Ge/subComponent1/IsParallel = "true"
<table>
<thead>
<tr>
<th>Ge/subComponent2/Type</th>
<th>&quot;TsBox&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/subComponent2/Parent</td>
<td>&quot;VrtParallelWorld&quot;</td>
</tr>
<tr>
<td>Ge/subComponent2/HLX</td>
<td>Ge/Wall/HLX m</td>
</tr>
<tr>
<td>Ge/subComponent2/HLY</td>
<td>Ge/Wall/HLY m</td>
</tr>
<tr>
<td>Ge/subComponent2/HLZ</td>
<td>Ge/subComponent/Thickness cm</td>
</tr>
<tr>
<td>Ge/subComponent2/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent2/TransZ</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent2/TransZTmp</td>
<td>Ge/subComponent1/TransZ + Ge/subComponent1/HLZ cm</td>
</tr>
<tr>
<td>Ge/subComponent2/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent2/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent2/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent2/IsParallel</td>
<td>&quot;true&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ge/subComponent3/Type</th>
<th>&quot;TsBox&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/subComponent3/Parent</td>
<td>&quot;VrtParallelWorld&quot;</td>
</tr>
<tr>
<td>Ge/subComponent3/HLX</td>
<td>Ge/Wall/HLX m</td>
</tr>
<tr>
<td>Ge/subComponent3/HLY</td>
<td>Ge/Wall/HLY m</td>
</tr>
<tr>
<td>Ge/subComponent3/HLZ</td>
<td>Ge/subComponent/Thickness cm</td>
</tr>
<tr>
<td>Ge/subComponent3/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent3/TransZ</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent3/TransZTmp</td>
<td>Ge/subComponent2/TransZ + Ge/subComponent2/HLZ cm</td>
</tr>
<tr>
<td>Ge/subComponent3/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent3/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent3/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent3/IsParallel</td>
<td>&quot;true&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ge/subComponent4/Type</th>
<th>&quot;TsBox&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/subComponent4/Parent</td>
<td>&quot;VrtParallelWorld&quot;</td>
</tr>
<tr>
<td>Ge/subComponent4/HLX</td>
<td>Ge/Wall/HLX m</td>
</tr>
<tr>
<td>Ge/subComponent4/HLY</td>
<td>Ge/Wall/HLY m</td>
</tr>
<tr>
<td>Ge/subComponent4/HLZ</td>
<td>Ge/subComponent/Thickness cm</td>
</tr>
<tr>
<td>Ge/subComponent4/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent4/TransZ</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent4/TransZTmp</td>
<td>Ge/subComponent3/TransZ + Ge/subComponent3/HLZ cm</td>
</tr>
<tr>
<td>Ge/subComponent4/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent4/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent4/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent4/IsParallel</td>
<td>&quot;true&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ge/subComponent5/Type</th>
<th>&quot;TsBox&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ge/subComponent5/Parent</td>
<td>&quot;VrtParallelWorld&quot;</td>
</tr>
<tr>
<td>Ge/subComponent5/HLX</td>
<td>Ge/Wall/HLX m</td>
</tr>
<tr>
<td>Ge/subComponent5/HLY</td>
<td>Ge/Wall/HLY m</td>
</tr>
<tr>
<td>Ge/subComponent5/HLZ</td>
<td>Ge/subComponent/Thickness cm</td>
</tr>
<tr>
<td>Ge/subComponent5/TransX</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent5/TransZ</td>
<td>0 cm</td>
</tr>
<tr>
<td>Ge/subComponent5/TransZTmp</td>
<td>Ge/subComponent4/TransZ + Ge/subComponent4/HLZ cm</td>
</tr>
<tr>
<td>Ge/subComponent5/RotX</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent5/RotY</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent5/RotZ</td>
<td>0 deg</td>
</tr>
<tr>
<td>Ge/subComponent5/IsParallel</td>
<td>&quot;true&quot;</td>
</tr>
</tbody>
</table>

| Ge/subComponent6/Type          | "TsBox"                     |
s:Ge/subComponent6/Parent = "VrtParallelWorld"
d:Ge/subComponent6/HLX = Ge/Wall/HLX m
d:Ge/subComponent6/HLY = Ge/Wall/HLY m
d:Ge/subComponent6/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent6/TransX = 0 cm
d:Ge/subComponent6/TransY = 0 cm
d:Ge/subComponent6/TransZ = Ge/subComponent6/TransZTmp + Ge/subComponent6/HLZ cm
d:Ge/subComponent6/RotX = 0 deg
d:Ge/subComponent6/RotY = 0 deg
d:Ge/subComponent6/RotZ = 0 deg
b:Ge/subComponent6/IsParallel = "true"

s:Ge/subComponent7/Type = "TsBox"
s:Ge/subComponent7/Parent = "VrtParallelWorld"
d:Ge/subComponent7/HLX = Ge/Wall/HLX m
d:Ge/subComponent7/HLY = Ge/Wall/HLY m
d:Ge/subComponent7/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent7/TransX = 0 cm
d:Ge/subComponent7/TransY = 0 cm
d:Ge/subComponent7/TransZ = Ge/subComponent7/TransZTmp + Ge/subComponent7/HLZ cm
d:Ge/subComponent7/RotX = 0 deg
d:Ge/subComponent7/RotY = 0 deg
d:Ge/subComponent7/RotZ = 0 deg
b:Ge/subComponent7/IsParallel = "true"

s:Ge/subComponent8/Type = "TsBox"
s:Ge/subComponent8/Parent = "VrtParallelWorld"
d:Ge/subComponent8/HLX = Ge/Wall/HLX m
d:Ge/subComponent8/HLY = Ge/Wall/HLY m
d:Ge/subComponent8/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent8/TransX = 0 cm
d:Ge/subComponent8/TransY = 0 cm
d:Ge/subComponent8/TransZ = Ge/subComponent8/TransZTmp + Ge/subComponent8/HLZ cm
d:Ge/subComponent8/RotX = 0 deg
d:Ge/subComponent8/RotY = 0 deg
d:Ge/subComponent8/RotZ = 0 deg
b:Ge/subComponent8/IsParallel = "true"

s:Ge/subComponent9/Type = "TsBox"
s:Ge/subComponent9/Parent = "VrtParallelWorld"
d:Ge/subComponent9/HLX = Ge/Wall/HLX m
d:Ge/subComponent9/HLY = Ge/Wall/HLY m
d:Ge/subComponent9/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent9/TransX = 0 cm
d:Ge/subComponent9/TransY = 0 cm
d:Ge/subComponent9/TransZ = Ge/subComponent9/TransZTmp + Ge/subComponent9/HLZ cm
d:Ge/subComponent9/RotX = 0 deg
d:Ge/subComponent9/RotY = 0 deg
d:Ge/subComponent9/RotZ = 0 deg
b:Ge/subComponent9/IsParallel = "true"

s:Ge/subComponent10/Type = "TsBox"
s:Ge/subComponent10/Parent = "VrtParallelWorld"
d:Ge/subComponent10/HLX = Ge/Wall/HLX m
TOPAS Documentation, Release 3.0

d:Ge/subComponent10/HLY = Ge/Wall/HLY m
d:Ge/subComponent10/HLZ = 0.5 * Ge/subComponent/Thickness cm
d:Ge/subComponent10/TransX = 0 cm
d:Ge/subComponent10/TransY = 0 cm
d:Ge/subComponent10/TransZTmp = Ge/subComponent9/TransZ + Ge/subComponent9/HLZ cm
d:Ge/subComponent10/TransZ = Ge/subComponent10/TransZTmp + Ge/subComponent10/HLZ cm
d:Ge/subComponent10/RotX = 0 deg
d:Ge/subComponent10/RotY = 0 deg
d:Ge/subComponent10/RotZ = 0 deg
b:Ge/subComponent10/IsParallel = "true"

#########################
# VRT
#########################
b:Vr/UseVarianceReduction = "true"
b:Vr/ParticleSplit/Active = "true"
sv:Vr/ParticleSplit/ParticleName = 1 "neutron"
s:Vr/ParticleSplit/Component = "VrtParallelWorld"
sv:Vr/ParticleSplit/SubComponents = 10
"subComponent1" "subComponent2"
"subComponent3" "subComponent4"
"subComponent5" "subComponent6"
"subComponent7" "subComponent8"
"subComponent9" "subComponent10"
s:Vr/ParticleSplit/Type = "ImportanceSampling"
sv:Vr/ParticleSplit/ImportanceValues = 10 2 4 8 16 32 64 128 256 512 1024

###############################################
# Scorer
###############################################
s:Sc/scorer/Quantity = "DoseToMedium"
s:Sc/scorer/Component = "Wall"
s:Sc/scorer/OutputFile = "ImportanceSampling"
s:Sc/scorer/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/scorer/Report = 3 "Sum" "Mean" "Variance"

###############################################
# Beam setting
###############################################
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "neutron"
s:So/Example/BeamEnergy = 10 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
s:So/Example/BeamPositionCutoffX = 10. cm
s:So/Example/BeamPositionCutoffY = 10. cm
s:So/Example/BeamPositionSpreadX = 0.65 cm
s:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
s:So/Example/BeamAngularCutoffX = 90. deg
s:So/Example/BeamAngularCutoffY = 90. deg
s:So/Example/BeamAngularSpreadX = 0.0032 rad
s:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10
:Ge/BeamPosition/TransZ = -1.0 * Ge/World/HLZ m
:Ge/BeamPosition/RotX = 0 deg

Ts/ShowHistoryCountAtInterval = 10

Ts/PauseBeforeQuit = "True"
Gr/view/Type = "OpenGl"
Gr/view/IncludeAxes = "True"

ImportanceSamplingMassGeometry.txt

INSTANTANEOUS

# Simple shielding example
#
:Ge/World/HLX = 2 m
:Ge/World/HLY = 2 m
:Ge/World/HLZ = 2.1 m
#
# Scoring region
#
s:Ge/Scorer/Type = "TsBox"
s:Ge/Scorer/Parent = "World"
b:Ge/Scorer/IsParallel = "True"
d:Ge/Scorer/HLX = 1 m
d:Ge/Scorer/HLY = 1 m
d:Ge/Scorer/HLZ = 1 m
d:Ge/Scorer/TransX = 0 m
d:Ge/Scorer/TransY = 0 m
d:Ge/Scorer/TransZ = 0 m
d:Ge/Scorer/RotX = 0 deg
d:Ge/Scorer/RotY = 0 deg
d:Ge/Scorer/RotZ = 0 deg
s:Ge/Scorer/Color = "red"
i:Ge/Scorer/XBins = 1
i:Ge/Scorer/YBins = 1
i:Ge/Scorer/ZBins = 10
#
# Geometry for vrt
#
d:Ge/subComponent/Thickness = 20 cm
d:Ge/subComponent/StartPosition = -1.0 * Ge/Scorer/HLZ cm
s:Ge/subComponent1/Material = "G4_CONCRETE"
s:Ge/subComponent2/Material = "G4_CONCRETE"
s:Ge/subComponent3/Material = "G4_CONCRETE"
s:Ge/subComponent4/Material = "G4_CONCRETE"
s:Ge/subComponent5/Material = "G4_CONCRETE"
s:Ge/subComponent6/Material = "G4_CONCRETE"
s:Ge/subComponent7/Material = "G4_CONCRETE"
s:Ge/subComponent8/Material = "G4_CONCRETE"
s:Ge/subComponent9/Material = "G4_CONCRETE"
s:Ge/subComponent10/Material = "G4_CONCRETE"
Ge/subComponent1/Type = "TsBox"
Ge/subComponent1/Parent = "World"
Ge/subComponent1/HLX = Ge/Scorer/HLX m
Ge/subComponent1/HLY = Ge/Scorer/HLY m
Ge/subComponent1/HLZ = 0.5 * Ge/subComponent/Thickness cm
Ge/subComponent1/TransX = 0 cm
Ge/subComponent1/TransY = 0 cm
Ge/subComponent1/TransZ = Ge/subComponent/StartPosition + Ge/subComponent1/HLZ cm
Ge/subComponent1/RotX = 0 deg
Ge/subComponent1/RotY = 0 deg
Ge/subComponent1/RotZ = 0 deg

Ge/subComponent2/Type = "TsBox"
Ge/subComponent2/Parent = "World"
Ge/subComponent2/HLX = Ge/Scorer/HLX m
Ge/subComponent2/HLY = Ge/Scorer/HLY m
Ge/subComponent2/HLZ = 0.5 * Ge/subComponent/Thickness cm
Ge/subComponent2/TransX = 0 cm
Ge/subComponent2/TransY = 0 cm
Ge/subComponent2/TransZ = Ge/subComponent2/TransZ + Ge/subComponent1/HLZ cm
Ge/subComponent2/RotX = 0 deg
Ge/subComponent2/RotY = 0 deg
Ge/subComponent2/RotZ = 0 deg

Ge/subComponent3/Type = "TsBox"
Ge/subComponent3/Parent = "World"
Ge/subComponent3/HLX = Ge/Scorer/HLX m
Ge/subComponent3/HLY = Ge/Scorer/HLY m
Ge/subComponent3/HLZ = 0.5 * Ge/subComponent/Thickness cm
Ge/subComponent3/TransX = 0 cm
Ge/subComponent3/TransY = 0 cm
Ge/subComponent3/TransZ = Ge/subComponent3/TransZ + Ge/subComponent2/HLZ cm
Ge/subComponent3/RotX = 0 deg
Ge/subComponent3/RotY = 0 deg
Ge/subComponent3/RotZ = 0 deg

Ge/subComponent4/Type = "TsBox"
Ge/subComponent4/Parent = "World"
Ge/subComponent4/HLX = Ge/Scorer/HLX m
Ge/subComponent4/HLY = Ge/Scorer/HLY m
Ge/subComponent4/HLZ = 0.5 * Ge/subComponent/Thickness cm
Ge/subComponent4/TransX = 0 cm
Ge/subComponent4/TransY = 0 cm
Ge/subComponent4/TransZ = Ge/subComponent4/TransZ + Ge/subComponent3/HLZ cm
Ge/subComponent4/RotX = 0 deg
Ge/subComponent4/RotY = 0 deg
Ge/subComponent4/RotZ = 0 deg

Ge/subComponent5/Type = "TsBox"
Ge/subComponent5/Parent = "World"
Ge/subComponent5/HLX = Ge/Scorer/HLX m
Ge/subComponent5/HLY = Ge/Scorer/HLY m
Ge/subComponent5/HLZ = 0.5 * Ge/subComponent/Thickness cm
Ge/subComponent5/TransX = 0 cm
Ge/subComponent5/TransY = 0 cm
d: Ge/subComponent5/Trans2Tmp = Ge/subComponent4/Trans2 + Ge/subComponent4/HLZ cm

d: Ge/subComponent5/Trans2 = Ge/subComponent5/Trans2Tmp + Ge/subComponent5/HLZ cm

d: Ge/subComponent5/RotX = 0 deg

d: Ge/subComponent5/RotY = 0 deg

d: Ge/subComponent5/RotZ = 0 deg

s: Ge/subComponent6/Type = "TsBox"
s: Ge/subComponent6/Parent = "World"
d: Ge/subComponent6/HLX = Ge/Scorer/HLX m
d: Ge/subComponent6/HLY = Ge/Scorer/HLY m
d: Ge/subComponent6/HLZ = 0.5 * Ge/subComponent/Thickness cm
d: Ge/subComponent6/TransX = 0 cm
d: Ge/subComponent6/TransY = 0 cm
d: Ge/subComponent6/TransZTmp = Ge/subComponent5/TransZ + Ge/subComponent5/HLZ cm
d: Ge/subComponent6/TransZ = Ge/subComponent6/TransZTmp + Ge/subComponent6/HLZ cm
d: Ge/subComponent6/RotX = 0 deg

d: Ge/subComponent6/RotY = 0 deg

d: Ge/subComponent6/RotZ = 0 deg

s: Ge/subComponent7/Type = "TsBox"
s: Ge/subComponent7/Parent = "World"
d: Ge/subComponent7/HLX = Ge/Scorer/HLX m
d: Ge/subComponent7/HLY = Ge/Scorer/HLY m
d: Ge/subComponent7/HLZ = 0.5 * Ge/subComponent/Thickness cm
d: Ge/subComponent7/TransX = 0 cm
d: Ge/subComponent7/TransY = 0 cm
d: Ge/subComponent7/TransZTmp = Ge/subComponent6/TransZ + Ge/subComponent6/HLZ cm
d: Ge/subComponent7/TransZ = Ge/subComponent7/TransZTmp + Ge/subComponent7/HLZ cm
d: Ge/subComponent7/RotX = 0 deg

d: Ge/subComponent7/RotY = 0 deg

d: Ge/subComponent7/RotZ = 0 deg

s: Ge/subComponent8/Type = "TsBox"
s: Ge/subComponent8/Parent = "World"
d: Ge/subComponent8/HLX = Ge/Scorer/HLX m
d: Ge/subComponent8/HLY = Ge/Scorer/HLY m
d: Ge/subComponent8/HLZ = 0.5 * Ge/subComponent/Thickness cm
d: Ge/subComponent8/TransX = 0 cm
d: Ge/subComponent8/TransY = 0 cm
d: Ge/subComponent8/TransZTmp = Ge/subComponent7/TransZ + Ge/subComponent7/HLZ cm
d: Ge/subComponent8/TransZ = Ge/subComponent8/TransZTmp + Ge/subComponent8/HLZ cm
d: Ge/subComponent8/RotX = 0 deg

d: Ge/subComponent8/RotY = 0 deg

d: Ge/subComponent8/RotZ = 0 deg

s: Ge/subComponent9/Type = "TsBox"
s: Ge/subComponent9/Parent = "World"
d: Ge/subComponent9/HLX = Ge/Scorer/HLX m
d: Ge/subComponent9/HLY = Ge/Scorer/HLY m
d: Ge/subComponent9/HLZ = 0.5 * Ge/subComponent/Thickness cm
d: Ge/subComponent9/TransX = 0 cm
d: Ge/subComponent9/TransY = 0 cm
d: Ge/subComponent9/TransZTmp = Ge/subComponent8/TransZ + Ge/subComponent8/HLZ cm
d: Ge/subComponent9/TransZ = Ge/subComponent9/TransZTmp + Ge/subComponent9/HLZ cm
d: Ge/subComponent9/RotX = 0 deg

d: Ge/subComponent9/RotY = 0 deg

d: Ge/subComponent9/RotZ = 0 deg
x:Ge/subComponent10/Type = "TsBox"
x:Ge/subComponent10/Parent = "World"
dx:Ge/subComponent10/HLX = Ge/Scorer/HLX m
dx:Ge/subComponent10/HLY = Ge/Scorer/HLY m
dx:Ge/subComponent10/HLZ = 0.5 * Ge/subComponent/Thickness cm
dx:Ge/subComponent10/TransX = 0 cm
dx:Ge/subComponent10/TransY = 0 cm
dx:Ge/subComponent10/TransZTmp = Ge/subComponent9/TransZ + Ge/subComponent9/HLZ cm
dx:Ge/subComponent10/TransZ = Ge/subComponent10/TransZTmp + Ge/subComponent10/HLZ cm
dx:Ge/subComponent10/RotX = 0 deg
dx:Ge/subComponent10/RotY = 0 deg
dx:Ge/subComponent10/RotZ = 0 deg

# VRT
b:Vr/UseVarianceReduction = "true"
b:Vr/ParticleSplit/Active = "true"
sv:Vr/ParticleSplit/ParticleName = 4 "neutron" "gamma" "proton" "e-"
sv:Vr/ParticleSplit/Component = "subComponent1"
sv:Vr/ParticleSplit/SubComponents = 9 "subComponent2" "subComponent3" "subComponent4" "subComponent5" "subComponent6" "subComponent7" "subComponent8" "subComponent9" "subComponent10"
s:Vr/ParticleSplit/Type = "ImportanceSampling"
s:Vr/ParticleSplit/ImportanceValues = 9 2 4 8 16 32 64 128 256 512

# Scorer

# Beam setting

# Beam setting

SecondaryBiasing.txt

# Secondary biasing occurs in the region
# of this component
s:Ge/target/Parent = "World"
s:Ge/target/Material = "Lead"
s:Ge/target/Type = "TsBox"
d:Ge/target/HLX = 1.5 cm
d:Ge/target/HLY = 1.5 cm
d:Ge/target/HLZ = 0.5 cm
s:Ge/target/Color = "yellow"
# Set the region
s:Ge/target/AssignToRegionNaMed = "targetregion"

# Component for scoring
s:Ge/scorer/Parent = "World"
s:Ge/scorer/Material = "G4_WATER"
s:Ge/scorer/Type = "TsCylinder"
d:Ge/scorer/RMax = 5 cm
d:Ge/scorer/HL = 0.05 mm
d:Ge/scorer/SPhi = 0 deg
d:Ge/scorer/DPhi = 360 deg
d:Ge/scorer/TransZ = 1 m + Ge/scorer/HL
d:Ge/scorer/Color = "yellow"
i:Ge/scorer/RBins = 20
i:Ge/scorer/PhiBins = 1
i:Ge/scorer/ZBins = 1
s:Sc/scorer/Quantity = "EnergyFluence"
s:Sc/scorer/Component = "scorer"
s:Sc/scorer/OutputFile = "secondaryBiasing"
s:Sc/scorer/IfOutputFileAlreadyExists = "Overwrite"

# VISUALIZATION
s:Gr/view/Type = "OpenGl"
i:Gr/view/WindowSizeX = 600
i:Gr/view/WindowSizeY = 600
d:Gr/view/Phi = 0 deg
d:Gr/view/Theta = 270 deg
u:Gr/view/Zoom = 5
b:Ts/PauseBeforeQuit = "True"
# SOURCE
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "e-"
d:So/Example/BeamEnergy = 6 MeV
d:So/Example/BeamEnergySpread = 0.1
d:So/Example/BeamPositionDistribution = "None"
d:So/Example/BeamAngularDistribution = "Gaussian"
d:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad

Ge/BeamPosition/Parent = "World"
Ge/BeamPosition/TransX = 0.0 cm
Ge/BeamPosition/TransZ = -1.0 * Ge/target/HLZ cm
Ge/BeamPosition/TransY = 0.0 mm
Ge/BeamPosition/RotX = 0 deg

# GEOMETRY
d:Ge/World/HLX = 16.1 cm
d:Ge/World/HLY = 16.1 cm
d:Ge/World/HLZ = 1.5 m
s:Ge/World/Material = "Air"
s:Ph/Default/Modules = 1 "g4em-standard_opt3"
i:So/Example/NumberOfHistoriesInRun = 1000
i:Ts/ShowHistoryCountAtInterval = 100
b:Ts/ShowCPUTime = "True"

#Secondary biasing
b:Vr/UseVarianceReduction = "True"
b:Vr/ParticleSplit/Active = "True"
s:Vr/ParticleSplit/Type = "secondarybiasing"
# The secondary particles produced in this process will
# be split 100 times if their energies are below of 6 MeV
sv:Vr/ParticleSplit/forregion/targetregion/processesNamed = 1 "eBrem"
uv:Vr/ParticleSplit/forRegion/targetregion/SplitNumber = 1 100.
dv:Vr/ParticleSplit/forRegion/targetregion/MaximumEnergies = 1 6 MeV

# Component that holds the reference frame for
# Directional split filter
s:Vr/ParticleSplit/ReferenceComponent = "target"
dv:Vr/ParticleSplit/forRegion/targetregion/DirectionalSplitLimits = 1 100 cm
dv:Vr/ParticleSplit/forRegion/targetregion/DirectionalSplitRadius = 1 5 cm

WeightWindow.txt

!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Simple shielding example
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
d:Ge/World/HLX = 2 m
d:Ge/World/HLY = 2 m
\begin{verbatim}
&: Ge/World/HLZ = 2.1 m

&: Ge/wall/Type = "TsBox"
&: Ge/wall/Parent = "World"
&: Ge/wall/Material = "G4_CONCRETE"
&: Ge/wall/HLX = 1 m
&: Ge/wall/HLY = 1 m
&: Ge/wall/HLZ = 1 m
&: Ge/wall/TransX = 0 m
&: Ge/wall/TransY = 0 m
&: Ge/wall/TransZ = 0 m
&: Ge/wall/RotX = 0 deg
&: Ge/wall/RotY = 0 deg
&: Ge/wall/RotZ = 0 deg
&: Ge/wall/Color = "red"
&: Ge/wall/XBins = 1
&: Ge/wall/YBins = 1
&: Ge/wall/ZBins = 10

# Geometry for vrt

&: Ge/subComponent/Thickness = 20 cm
&: Ge/subComponent/StartPosition = -1.0 * Ge/wall/HLZ cm

&: Ge/VrtParallelWorld/Type = "TsBox"
&: Ge/VrtParallelWorld/Parent = "World"
&: Ge/VrtParallelWorld/HLX = 1.1 m
&: Ge/VrtParallelWorld/HLY = 1.1 m
&: Ge/VrtParallelWorld/HLZ = 1.1 m
&: Ge/VrtParallelWorld/TransX = 0 cm
&: Ge/VrtParallelWorld/TransY = 0 cm
&: Ge/VrtParallelWorld/TransZ = 0 cm
&: Ge/VrtParallelWorld/RotX = 0 deg
&: Ge/VrtParallelWorld/RotY = 0 deg
&: Ge/VrtParallelWorld/RotZ = 0 deg
&: Ge/VrtParallelWorld/IsParallel = "true"

&: Ge/subComponent1/Type = "TsBox"
&: Ge/subComponent1/Parent = "VrtParallelWorld"
&: Ge/subComponent1/HLX = Ge/Wall/HLX m
&: Ge/subComponent1/HLY = Ge/Wall/HLY m
&: Ge/subComponent1/HLZ = 0.5 * Ge/subComponent/Thickness cm
&: Ge/subComponent1/TransX = 0 cm
&: Ge/subComponent1/TransY = 0 cm
&: Ge/subComponent1/TransZ = Ge/subComponent/StartPosition + Ge/subComponent1/HLZ cm
&: Ge/subComponent1/RotX = 0 deg
&: Ge/subComponent1/RotY = 0 deg
&: Ge/subComponent1/RotZ = 0 deg
&: Ge/subComponent1/IsParallel = "true"

&: Ge/subComponent2/Type = "TsBox"
&: Ge/subComponent2/Parent = "VrtParallelWorld"
&: Ge/subComponent2/HLX = Ge/Wall/HLX m
&: Ge/subComponent2/HLY = Ge/Wall/HLY m
&: Ge/subComponent2/HLZ = 0.5 * Ge/subComponent/Thickness cm
&: Ge/subComponent2/TransX = 0 cm
\end{verbatim}
\[
\text{Ge/subComponent6/TransZ} = \text{Ge/subComponent6/TransZ} + \text{Ge/subComponent6/HLZ} \text{ cm}
\]
\[
\text{Ge/subComponent6/RotX} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent6/RotY} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent6/RotZ} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent6/IsParallel} = "true"
\]
\[
\text{Ge/subComponent7/Type} = "TsBox"
\]
\[
\text{Ge/subComponent7/Parent} = "VrtParallelWorld"
\]
\[
\text{Ge/subComponent7/HLX} = \text{Ge/Wall/HLX} \text{ m}
\]
\[
\text{Ge/subComponent7/HLY} = \text{Ge/Wall/HLY} \text{ m}
\]
\[
\text{Ge/subComponent7/HLZ} = 0.5 \times \text{Ge/subComponent/Thickness} \text{ cm}
\]
\[
\text{Ge/subComponent7/TransX} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent7/TransY} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent7/TransZ} = \text{Ge/subComponent6/TransZ} + \text{Ge/subComponent6/HLZ} \text{ cm}
\]
\[
\text{Ge/subComponent7/RotX} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent7/RotY} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent7/RotZ} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent7/IsParallel} = "true"
\]
\[
\text{Ge/subComponent8/Type} = "TsBox"
\]
\[
\text{Ge/subComponent8/Parent} = "VrtParallelWorld"
\]
\[
\text{Ge/subComponent8/HLX} = \text{Ge/Wall/HLX} \text{ m}
\]
\[
\text{Ge/subComponent8/HLY} = \text{Ge/Wall/HLY} \text{ m}
\]
\[
\text{Ge/subComponent8/HLZ} = 0.5 \times \text{Ge/subComponent/Thickness} \text{ cm}
\]
\[
\text{Ge/subComponent8/TransX} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent8/TransY} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent8/TransZ} = \text{Ge/subComponent7/TransZ} + \text{Ge/subComponent7/HLZ} \text{ cm}
\]
\[
\text{Ge/subComponent8/RotX} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent8/RotY} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent8/RotZ} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent8/IsParallel} = "true"
\]
\[
\text{Ge/subComponent9/Type} = "TsBox"
\]
\[
\text{Ge/subComponent9/Parent} = "VrtParallelWorld"
\]
\[
\text{Ge/subComponent9/HLX} = \text{Ge/Wall/HLX} \text{ m}
\]
\[
\text{Ge/subComponent9/HLY} = \text{Ge/Wall/HLY} \text{ m}
\]
\[
\text{Ge/subComponent9/HLZ} = 0.5 \times \text{Ge/subComponent/Thickness} \text{ cm}
\]
\[
\text{Ge/subComponent9/TransX} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent9/TransY} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent9/TransZ} = \text{Ge/subComponent8/TransZ} + \text{Ge/subComponent8/HLZ} \text{ cm}
\]
\[
\text{Ge/subComponent9/RotX} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent9/RotY} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent9/RotZ} = 0 \text{ deg}
\]
\[
\text{Ge/subComponent9/IsParallel} = "true"
\]
\[
\text{Ge/subComponent10/Type} = "TsBox"
\]
\[
\text{Ge/subComponent10/Parent} = "VrtParallelWorld"
\]
\[
\text{Ge/subComponent10/HLX} = \text{Ge/Wall/HLX} \text{ m}
\]
\[
\text{Ge/subComponent10/HLY} = \text{Ge/Wall/HLY} \text{ m}
\]
\[
\text{Ge/subComponent10/HLZ} = 0.5 \times \text{Ge/subComponent/Thickness} \text{ cm}
\]
\[
\text{Ge/subComponent10/TransX} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent10/TransY} = 0 \text{ cm}
\]
\[
\text{Ge/subComponent10/TransZ} = \text{Ge/subComponent9/TransZ} + \text{Ge/subComponent9/HLZ} \text{ cm}
\]
\[
\text{Ge/subComponent10/TransZ} = \text{Ge/subComponent10/TransZ} + \text{Ge/subComponent10/HLZ} \text{ cm}
\]
d:Ge/subComponent10/RotX = 0 deg
d:Ge/subComponent10/RotY = 0 deg
d:Ge/subComponent10/RotZ = 0 deg
b:Ge/subComponent10/IsParallel = "true"

#########################
# VRT
#########################
b:Vr/UseVarianceReduction = "true"
b:Vr/ParticleSplit/Active = "true"
sv:Vr/ParticleSplit/ParticleName = 1 "neutron"
s:Vr/ParticleSplit/Component = "VrtParallelWorld"
sv:Vr/ParticleSplit/SubComponents = 10
"subComponent1" "subComponent2"
"subComponent3" "subComponent4"
"subComponent5" "subComponent6"
"subComponent7" "subComponent8"
"subComponent9" "subComponent10"
s:Vr/ParticleSplit/Type = "WeightWindow"

uv:Vr/ParticleSplit/WeightMap = 10 1.0 0.5 0.25 0.125 0.0625 0.03125 0.015625 0.0078125 0.00390625 0.001953125

uv:Vr/ParticleSplit/EnergyMap = 10 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 GeV

u:Vr/ParticleSplit/UpperLimitFactor = 1
u:Vr/ParticleSplit/SurvivalFactor = 1
i:Vr/ParticleSplit/MaximumSplitNumber = 100
s:Vr/ParticleSplit/PlaceOfAction = "OnBoundary"

#########################
# Scorer
#########################
s:Sc/scorer/Quantity = "DoseToMedium"
s:Sc/scorer/Component = "Wall"
s:Sc/scorer/OutputFile = "WeightWindow"
s:Sc/scorer/IfOutputFileAlreadyExists = "Overwrite"
sv:Sc/scorer/Report = 3 "Sum" "Mean" "Variance"

#########################
# Beam setting
#########################
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
s:So/Example/BeamParticle = "neutron"
d:So/Example/BeamEnergy = 10 MeV
u:So/Example/BeamEnergySpread = 0.757504
s:So/Example/BeamPositionDistribution = "Gaussian"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
s:So/Example/BeamPositionCutoffX = 10. cm
d:So/Example/BeamPositionCutoffY = 10. cm
d:So/Example/BeamPositionSpreadX = 0.65 cm
d:So/Example/BeamPositionSpreadY = 0.65 cm
s:So/Example/BeamAngularDistribution = "Gaussian"
s:So/Example/BeamAngularCutoffX = 90. deg
d:So/Example/BeamAngularCutoffY = 90. deg
d:So/Example/BeamAngularSpreadX = 0.0032 rad
d:So/Example/BeamAngularSpreadY = 0.0032 rad
i:So/Example/NumberOfHistoriesInRun = 10

d:Ge/BeamPosition/TransZ = -1.0 * Ge/World/HLZ m
d:Ce/BeamPosition/RotX = 0 deg
i:Ts/ShowHistoryCountAtInterval = 1
b:Ts/PauseBeforeQuit = "True"
s:Gr/view/Type = "OpenGL"
b:Gr/view/IncludeAxes = "True"
PlasticScintillator.txt

# Demonstrates optical behavior in a plastic scintillator

includeFile = OpticalMaterialSample.txt

i:Ts/SequenceVerbosity = 1

# SEQUENCE
b:Ts/PauseBeforeQuit = "True"
b:Ts/ShowCPUtime = "True"
i:Ts/ShowHistoryCountAtInterval = 1

# SCORER
s:Sc/PhSp/Quantity = "PhaseSpace"
s:Sc/PhSp/Surface = "Plastic/OuterCurvedSurface"
s:Sc/PhSp/OutputType = "ASCII"
s:Sc/PhSp/OutputFile = "PHSP"
s:Sc/PhSp/IfOutputFileAlreadyExists = "Overwrite"
b:Sc/PhSp/IncludeTimeOfFlight = "true"
b:Sc/PhSp/UsePDGEncoding = "True"

# VISUALIZATION
s:Gr/view/Type = "OpenGl"
i:Gr/view/WindowSizeX = 600
i:Gr/view/WindowSizeY = 600
d:Gr/view/Theta = 45. deg
d:Gr/view/Phi = 45. deg

# PHYSICS LIST
s:Ph/ListName = "Optical"
s:Ph/Optical/Type = "Geant4_Modular"
sv:Ph/Optical/Modules = 2 "g4em-standard_opt3" "g4optical"
# SOURCE
s:So/Example/Type = "Beam"
s:So/Example/Component = "BeamPosition"
d:So/Example/BeamParticle = "proton"
#:So/Example/BeamEnergy Spread = 0.5
s:So/Example/BeamEnergy = 100 MeV
s:So/Example/BeamPositionDistribution = "Flat"
s:So/Example/BeamPositionCutoffShape = "Ellipse"
d:So/Example/BeamPositionCutoffX = 0.5 mm
d:So/Example/BeamPositionCutoffY = 0.5 mm
s:So/Example/BeamAngularDistribution = "Flat"
d:So/Example/BeamAngularCutoffX = 0.2 rad
d:So/Example/BeamAngularCutoffY = 0.2 rad
i:So/Example/NumberOfHistoriesInRun = 1

# GEOMETRY
d:Ge/World/HLX = 1 cm
d:Ge/World/HLY = 1 cm
d:Ge/World/HLZ = 1 cm
s:Ge/World/Material = "Air"

# Plastic
ds:Ge/Plastic/Parent = "World"
s:Ge/Plastic/Type = "TsCylinder"
s:Ge/Plastic/Material = "BuapCfm"
d:Ge/Plastic/HL = 0.5 cm
d:Ge/Plastic/RMin = 0.0 cm
d:Ge/Plastic/RMax = 0.5 cm
d:Ge/Plastic/SPhi = 0 deg
d:Ge/Plastic/DPhi = 360 deg
d:Ge/Plastic/TransX = 0 mm
d:Ge/Plastic/TransY = 0 mm
d:Ge/Plastic/TransZ = 0 mm
d:Ge/Plastic/RotX = 0 deg
d:Ge/Plastic/RotY = 0 deg
d:Ge/Plastic/RotZ = 0 deg

Rotating_Surfaces.txt

# Demonstrates optical behavior on rotating surfaces
includeFile = Surfaces.txt
i:Tf/SequenceVerbosity = 0
i:Tf/Verbosity = 0
d:Tf/TimelineEnd = 60.0 ms
i:Tf/NumberOfSequentialTimes = 40
i:So/Example/NumberOfHistoriesInRun = 20
d:Ge/Splitter/RotX = Tf/SplitterRot/Value deg
d:Tf/SplitterRot/Function = "Linear deg"
Surfaces.txt

# Demonstrate use of optical surfaces

includeFile = OpticalMaterialSample.txt

Ts/SequenceVerbosity = 1

# SEQUENCE
Ts/PauseBeforeQuit = "True"
Ts/ShowCPUTime = "True"
Ts/ShowHistoryCountAtInterval = 10

# VISUALIZATION
Gr/view/Type = "OpenGL"
# Gr/view/IncludeAxes = "true"
Gr/view/AxesSize = 10 cm

# PHYSICS LIST
Ph/ListName = "Optical"
Ph/Optical/Type = "Geant4_Modular"
Ph/Optical/Modules = 2 "g4optical" "g4em-standard_opt3"

# SCORER
Sc/PhSp/Quantity = "PhaseSpace"
Sc/PhSp/Surface = "Screen/YMinusSurface"
Sc/PhSp/OutputType = "ASCII"
Sc/PhSp/OutputFile = "PhSp"
Sc/PhSp/IfOutputFileAlreadyExists = "Overwrite"
Sc/PhSp/UsePDGEncoding = "True"
Sc/PhSp/KillAfterPhaseSpace = "True"

# SOURCE
So/Example/Type = "Beam"
So/Example/Component = "BeamPosition"
So/Example/BeamParticle = "opticalphoton"
So/Example/BeamPolarizationX = 1
So/Example/BeamPolarizationY = 0
So/Example/BeamPolarizationZ = 0
So/Example/BeamEnergy = 1.9593 eV
So/Example/BeamEnergySpread = 0.04898
So/Example/BeamPositionDistribution = "Flat"
So/Example/BeamPositionCutoffShape = "Ellipse"
So/Example/BeamPositionCutoffX = 0.405 mm
...
WavelengthShifter.txt

# Demonstrates optical behavior in a wavelength shifter

includeFile = OpticalMaterialSample.txt

ts:SequenceVerbosity = 0

# SEQUENCE
b:PauseBeforeQuit = "False"
b:ShowCPUtime = "True"
ts:ShowHistoryCountAtInterval = 10000

# SCORER
s:PhSp1/Quantity = "PhaseSpace"
s:PhSp1/Surface = "WLS/OuterCurvedSurface"
s:PhSp1/OutputType = "ASCII"
s:PhSp1/OutputFile = "In"
### Sc/PhSp1

- `IfOutputFileAlreadyExists = "Overwrite"`
- `IncludeTimeOfFlight = "true"
- `UsePDGEncoding = "True"
- `OnlyIncludeParticlesFromVolume = 1 "Plastic"
- `OnlyIncludeParticlesGoing = "in"

### Sc/PhSp2

- `Quantity = "PhaseSpace"
- `Surface = "WLS/Core/ZPlusSurface"
- `OutputType = "ASCII"
- `OutputFile = "Out"
- `IfOutputFileAlreadyExists = "Overwrite"
- `IncludeTimeOfFlight = "true"
- `UsePDGEncoding = "True"

### VISUALIZATION

- `view/Type = "OpenGl"
- `IncludeAxes = "True"
- `WindowSizeX = 600
- `WindowSizeY = 600
- `Theta = 45. deg
- `Phi = 45. deg

### PHYSICS LIST

- `ListName = "Optical"
- `Optical/Type = "Geant4_Modular"
- `Modules = 7 "g4optical" "g4em-standard_opt3" "g4-h-phy_QGSP_BIC_HP"
  - "g4decay"
  - "g4ion-binarycascade" "g4-elastic_HP" "g4stopping"

### SOURCE

- `Example/Type = "Beam"
- `Component = "BeamPosition"
- `BeamParticle = "proton"
- `BeamEnergy = 200 MeV
- `BeamEnergySpread = 0.5
- `BeamPositionDistribution = "Flat"
- `BeamPositionCutoffShape = "Ellipse"
- `BeamPositionCutoffX = 0.5 mm
- `BeamPositionCutoffY = 0.5 mm
- `BeamAngularDistribution = "Gaussian"
- `BeamAngularCutoffX = 90. deg
- `BeamAngularCutoffY = 90. deg
- `BeamAngularSpreadX = 0.2 rad
- `BeamAngularSpreadY = 0.2 rad
- `NumberOfHistoriesInRun = 100000
- `BeamPosition/TransY = -3.0 cm

### GEOMETRY

- `World/HLX = 6 cm
- `World/HLY = 6 cm
- `World/HLZ = 20 cm
- `World/Material = "Air"

### Plastic

- `WaterBox/Parent = "World"
- `WaterBox/Type = "TsBox"
- `WaterBox/Material = "G4_WATER"
- `WaterBox/HLX = 2.5 cm
<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_{\text{Ge/WaterBox/HLY}})</td>
<td>2.5 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/HLZ}})</td>
<td>5.0 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/TransX}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/TransY}})</td>
<td>-3.0 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/TransZ}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/RotX}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/RotY}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WaterBox/RotZ}})</td>
<td>0 deg</td>
</tr>
</tbody>
</table>

# Plastic

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_{\text{Ge/Plastic/Parent}})</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/Plastic/Type}})</td>
<td>&quot;TsBox&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/Plastic/Material}})</td>
<td>&quot;BUAPFCFM&quot;</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/HLX}})</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/HLY}})</td>
<td>0.5 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/HLZ}})</td>
<td>2.5 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/TransX}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/TransY}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/TransZ}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/RotX}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/RotY}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/Plastic/RotZ}})</td>
<td>0 deg</td>
</tr>
</tbody>
</table>

# WLS

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d_{\text{Ge/WLS/Diameter}})</td>
<td>1.03 mm</td>
</tr>
<tr>
<td>(s_{\text{Ge/WLS/Parent}})</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/WLS/Type}})</td>
<td>&quot;TsCylinder&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/WLS/Material}})</td>
<td>&quot;PMMA&quot;</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/HL}})</td>
<td>10 cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/RMin}})</td>
<td>0.50 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/RMax}})</td>
<td>0.515 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/SPhi}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/DPhi}})</td>
<td>360 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/TransX}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/TransY}})</td>
<td>Ge/Plastic/HLY + Ge/WLS/RMax mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/TransZ}})</td>
<td>Ge/WLS/HL - Ge/Plastic/HLZ cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/RotX}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/RotY}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/RotZ}})</td>
<td>0 deg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s_{\text{Ge/WLS/Core/Parent}})</td>
<td>&quot;World&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/WLS/Core/Type}})</td>
<td>&quot;TsCylinder&quot;</td>
</tr>
<tr>
<td>(s_{\text{Ge/WLS/Core/Material}})</td>
<td>&quot;Polystyrene&quot;</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/HL}})</td>
<td>Ge/WLS/HL cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/RMin}})</td>
<td>0.0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/RMax}})</td>
<td>0.5 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/SPhi}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/DPhi}})</td>
<td>360 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/TransX}})</td>
<td>0 mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/TransY}})</td>
<td>Ge/Plastic/HLY + Ge/WLS/RMax mm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/TransZ}})</td>
<td>Ge/WLS/HL - Ge/Plastic/HLZ cm</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/RotX}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/RotY}})</td>
<td>0 deg</td>
</tr>
<tr>
<td>(d_{\text{Ge/WLS/Core/RotZ}})</td>
<td>0 deg</td>
</tr>
</tbody>
</table>
While most TOPAS users will find that they can implement everything they want from parameter files, those who require additional functionality are free to write their own C++ code to extend TOPAS. Your code can take advantage of the full syntax richness of C++. You may use almost any Geant4 class in your work.

These new classes can be:

- Geometry Components
- Scorers
- Filters
- Physics Lists
- Particle Sources
- Magnetic Field Descriptions
- ElectroMagnetic Field Descriptions
- Imaging To Material Converters

And you can also provide classes that will be called to do whatever you want at:

- Start or End of Session
- Start or End of Run
- Start or End of History

As the first line of each of your classes, you will provide very specific comment lines that tells us how to weave your class into the rest of TOPAS. For example, to define your own Geometry Component, your class will start with something like:

```cpp
// Component for MyComponent1
```

This tells TOPAS that your class defines a Geometry Component, and that this component should be used if a parameter file has the matching component type:
C++ does not require that a given file, such as MyComponent1.cc, contain a class of the same name. However, the Topas make system DOES require that this file name and class name match. So, for example, a file named MyComponent1.cc and its corresponding MyComponent1.hh must contain a class named MyComponent1.

You can find a set of example extensions on the topasmc.org code repository page. You can see there what the special comment string is for each type of class (Geometry Component, Scorer, Filter, etc.).

To build your new TOPAS executable that incorporates all of your extensions, you run CMake with an argument that tells it the location of your extensions. Your extensions then coexist with the rest of the TOPAS code.

You can even have subdirectories within your extensions directory, so that you might for example have different subdirectories with extensions from different collaborators:

```plaintext
extensions/my_extensions_from_university_a
extensions/my_extensions_from_company_b
extensions/some_other_extensions
```

Our CMake script will recursively search your extensions directory to take all of your extensions.

Details are in the README in TOPAS.

Even when you have to write your own C++ code, TOPAS work is still easier than plain Geant4. You write your extensions as concrete implementations of TOPAS base classes which provide a wealth of helper functions to simplify your work. You may use the TOPAS parameter system to provide parameters to your classes, and those parameters can vary in time, like any other TOPAS parameters. All user extensions have a pointer to the parameter manager in their constructor. Thus, to access TOPAS parameters, call one of the following methods: `fPm->someMethod`

In all of the following forms, the `parameterName` argument can be either a `G4String` or a `char*`.

```cpp
// See if parameter exists
G4bool ParameterExists(parameterName);
// Get number of values in a vector parameter
G4int GetVectorLength(parameterName);
// Get dimensioned double value of parameter in Geant4's internal units
G4double GetDoubleParameter(parameterName, const char* unitCategory);
// Get double value of a unitless parameter
G4double GetUnitlessParameter(parameterName);
// Get integer value of parameter
G4int GetIntegerParameter(parameterName);
// Get Boolean value of parameter
G4bool GetBooleanParameter(parameterName);
// Get string value of parameter (whether it is actually a string parameter or not)
G4String GetStringValue(parameterName);
// Get vector of dimensioned double values of parameter in Geant4's internal units
G4double* GetDoubleVector(parameterName, const char* unitCategory);
// Get vector of double values of a unitless parameter
G4double* GetUnitlessVector(parameterName);
// Get vector of integer values of parameter
G4int* GetIntegerVector(parameterName);
// Get vector of Boolean values of parameter
G4bool* GetBooleanVector(parameterName);
// Get vector of string values of parameter
G4String* GetStringVector(parameterName);
// Get TwoVector of double values of parameter in Geant4's internal units
G4TwoVector GetTwoVectorParameter(parameterName, const char* unitCategory);
// Get ThreeVector of double values of parameter in Geant4's internal units
G4ThreeVector GetThreeVectorParameter(parameterName, const char* unitCategory);
```
Stubs of extension classes are included in the topas/extensions directory in your TOPAS release. A set of additional example components, scorers and filters are distributed as a zip file on the TOPAS web site (see the file called extension_examples...). To create your own extension, start with the example that is the closest to what you want, then change the file name (and the class name throughout the file), then adjust the code as you wish.

We believe this extensions mechanism should allow you to do almost anything you like from within TOPAS. If you find any significant limitations, please reach out to us. We want to enable your unique research.

**Extra Classes**

First line of the cc file must be of the form:

```cpp
// Extra Class for use by TsMyBeginHistory
```

Any of your extension classes are welcome to themselves instantiate other classes. You just need to advise us to link in these classes by providing the above special line.

**Changeable Parameters**

In general, parameters cannot change once the TOPAS session has begun. Changes due to Time Features are fine (since the time feature’s behavior itself is well defined), but any other change violates basic principles of repeatability.

C++ code that changes a parameter during the session, aside from time features, is allowed only for a special case in which a specialized geometry component needs to set a parameter value on the fly. An example is when TsCompensator reads in the compensator definition from a special file format. The resulting compensator thickness updates a parameter that affects positioning of other components.

Such a special case is allowed if the relevant parameter is defined from the start to be “Changeable”. This is done by adding a c at the end of the parameter type, for example:

```cpp
dc:Ge/Compensator/TransZ = 2. cm # the initial dc indicates that this is a double
  ←that is changeable
```

For vector parameters, the c still comes just before the colon, for example:

```cpp
gvc:...
```

In a complex parameter file chain, if any level of the chain redefines this as just a d rather than a dc, other parameter files will see this as a non-changeable parameter. Thus one parameter file may lock out others from making such changes.

TOPAS makes note of which parts of the system uses this changeable parameter (either directly or through a chain of parameters depending on other parameters) and takes care to explicitly update those parts of the system if this parameter ever changes.

Of course any parameter value can override the same parameter’s value from a parent parameter file. This override at initial parameter read-in time is not what we mean by changeable. By Changeable we mean a value that changes during the TOPAS session.

The c syntax is not required when you are simply setting a parameter’s value to a time feature. We allow:

```cpp
d:Ge/Propeller/RotZ = Tf/PropellerRot/Value
```

It is true that this Tf/PropellerRot/Value is changeable, but that is handled internally by TOPAS.
Transient Parameters

When a parameter is changed during the session, either because it is a time feature value, or because some piece of C++ code changes the parameter, TOPAS does not actually overwrite the original parameter in memory, but instead adds it to a “Transient Parameter List”. The Transient Parameter list always takes precedence over any other parameters file.

Transient parameters may be the first occurrence of a given parameter, as for the materials for a patient that are only instantiated as the patient is read in from DICOM, or transient parameters may override previously-defined parameters.
First line of the cc file must be of the form:

```cpp
// Component for MyComponent1
```

TOPAS geometry components are like small pieces of what Geant4 users call their “detector construction” class. The Geometry sections of the Geant4 Application Developers Guide provide details on the full geometrical functionality of Geant4. In this section, we explain some details about how to write TOPAS components, but we assume that you are already comfortable with basic concepts of C++ and Geant4 geometry. The notes below are intended to discuss only those parts which may not be obvious.

Your geometry component class will be a concrete implementation of the base class: TsVGeometryComponent.

You can get any parameter name of the current component by using the GetFullParmName method. For example, if your parameter file specifies:

```
d:Ge/MyComponent/Blatz = 42. mm
```

GetFullParmName("Blatz") will return Ge/MyComponent/Blatz. You can then feed this resulting string into the parameter access methods such as:

```cpp
g4double blatzLength = fPm->GetDoubleParameter(GetFullParmName("Blatz"), "length");
```

Your component may contain any of the following methods. Careful attention to what goes where will insure that your classes are robust under 4D and the base classes will do much of the work for you.

Constructor: Must exist and may be empty. This method will only be called at the very beginning of the simulation. It will not be called after changes in 4D. Only put things here if you are absolutely certain you will not need to recompute them during the simulation.

Destructor: Must exist and may be empty. Destroy any special objects you created with “new” statements. You may not destroy solids, logical volumes or physical volumes. These destructions are handled for you by the base class.

You do not need to do anything to handle the basic parameters, Parent, TransX, TransY, TransZ, RotX, RotY, RotZ, Material and Color. These are handled for you by the base class, including 4D capabilities.
If there are any other parameters that you may want to vary in 4D, provide a method `UpdateForSpecificParameterChange(G4String parameter)`.

- If the parameter name is one that you want to handle, do so. Be sure to use `GetFullParmNameLower` rather than `GetFullParmName` in your check.
- If your handling of this parameter moves a volume relative to its mother volume, advise Geant4’s smart voxel system that it needs to re-optimize the mother volume by calling `AddToReoptimizeList`. The argument should be the mother’s logical volume.
- If the parameter name is not one that you want to handle, pass it on to the base class handler, `TsVGeometryComponent::UpdateForSpecificParameterChange`. This is essential to enable basic 4D behaviors such as overall component motion.

`TsMultiLeafCollimator.cc` is a good example of this kind of behavior. It allows leaf position to change over time.

For the rest of your work, provide a method `Construct`.

The first line of `Construct` MUST be:

```cpp
BeginConstruction();
```

The rest of `Construct` is whatever you want to do to create Geant4 Solids, Logical Volumes and Physical Volumes. But you must follow some rules to insure that TOPAS will be able to properly manage your volumes in 4D.

- You create Geant4 Solids just as you would in any Geant4 geometry.
- You DO NOT create Geant4 Logical Volumes or Physical Volumes directly, but instead use helper methods from the base class. This allows TOPAS to manage your solids and volumes efficiently, even if they are moving.

To create the overall logical volume for your component, use:

```cpp
fEnvelopeLog = CreateLogicalVolume(G4VSolid* solid);
```

The logical volume will automatically get the material and visualization properties specified in your parameter file for this component, such as `Ge/MyComponent/Material` and `Ge/MyComponent/Color`. Be sure that the value on the left side of the above is exactly “fEnvelopeLog”. This insures that TOPAS knows the overall logical volume’s name and is essential for TOPAS to support your component in 4D.

If a component is made up of more than one volume, these additional volumes are called “SubComponents.” A component may have more zero, one or more SubComponents. An example of SubComponents is the Blades in a Propeller, such as:

```cpp
s:Ge/Propeller/Type = "TsPropeller"
... 
1:Ge/Propeller/NbOfBlades = Ge/PropellerConstant/NbBlades 
s:Ge/Propeller/Blade/Material = "World"
 s:Ge/Propeller/Blade/Color = "skyblue"
```

In all of the following forms, the subComponentName argument can be either a `G4String` or a `char*`.

To create a logical volume for a subcomponent, use:

```cpp
G4LogicalVolume* CreateLogicalVolume(subComponentName, G4VSolid* solid);
```

TOPAS will look for material and visualization parameters such as:

```cpp
Ge/ComponentName/SubComponentName/Material = ...
```

To hard-code the material, rather than having it come from this parameter, use:
This is particularly useful in cases where you want the material to be the same as the component’s mother, that is, the material surrounding your component. We do this, for example, when we want to make a void in a collimator. To get that mother volume’s material name, use:

```cpp
G4String envelopeMaterialName = fParentComponent->GetResolvedMaterialName();
```

The base class will take care of automatically setting your component’s visualization attributes based on the component’s parameters. But you can set different attributes for subComponents with code such as:

```cpp
G4VisAttributes* yokeColor = new G4VisAttributes(G4Colour(0.2, 1.0, 0.2)); // Sets RGB color
RegisterVisAtt(yokeColor); // Necessary so that TOPAS can delete the attribute if the component is rebuild during 4D behavior
yokeLogicalVolumePointer->SetVisAttributes(yokeColor);
```

To create the overall physical volume for your component, use:

```cpp
fEnvelopePhys = CreatePhysicalVolume(fEnvelopeLog);
```

Be sure that the value on the left side of the above is exactly `fEnvelopePhys`. This insures that TOPAS knows the overall physical volume’s name and is essential for TOPAS to support your component in 4D.

Additional forms of CreatePhysicalVolume allow you to place subcomponents within your component.

To place a subcomponent in the center of your logical volume `lVol`:

```cpp
G4VPhysicalVolume* CreatePhysicalVolume(subComponentName, G4LogicalVolume* lVol, G4VPhysicalVolume* parent);
```

To place a subcomponent into your logical volume `lVol`, with an offset or rotation:

```cpp
G4VPhysicalVolume* CreatePhysicalVolume(subComponentName, G4LogicalVolume* lVol, G4RotationMatrix* rot, G4ThreeVector* trans, G4VPhysicalVolume* parent);
```

To place multiple copies of the same subcomponent name into your logical volume, call:

```cpp
G4VPhysicalVolume* CreatePhysicalVolume(subComponentName, G4int copy, G4bool reuseLogical, G4LogicalVolume* lVol, G4RotationMatrix* rot, G4ThreeVector* trans, G4VPhysicalVolume* parent);
```

- `copy` should be a unique integer to differentiate the different copies of your subcomponent. This copy number is useful in some of the visualization commands when you want to control just one copy or another.
- Set `reuseLogical` true if you are using the same logical volume in all of these placements. This is efficient if all of the copies of the subcomponent are identical except for their placement.
- Set `reuseLogical` false if you are using different logical volumes in each of these placements. This allows you to make each copy of the subcomponent different (different material, different shape, different size, etc.).

To place multiple copies of the same subcomponent using a Geant4 parameterization (creating Geant4 parameterized volumes), call:

```cpp
G4VPhysicalVolume* CreatePhysicalVolume(const char* subComponentName, G4LogicalVolume* lVol, G4VPhysicalVolume* parent, const EAxis pAxis, const G4int nReplicas, G4VPVParameterisation* pParam);
```
To place multiple copies of the same subcomponent using a Geant4 replica volume, call:

```cpp
G4VPhysicalVolume* CreatePhysicalVolume(const char* subComponentName, G4LogicalVolume* lVol, G4VPhysicalVolume* parent, const EAxis pAxis, const G4int nReplicas, G4double width);
```

The last line of Construct MUST be:

```cpp
return fEnvelopePhys;
```

Some helper functions you may want to use from the TsParameterManager:

```cpp
G4VisAttributes* GetColor(G4String name);
G4VisAttributes* GetColor(const char* name);
G4VisAttributes* GetInvisible();
```

Some helper functions you may want to use from the TsVGeometryComponent:

```cpp
SetTooComplexForOGLS()
```

Call this to tell Graphics that this component has become too complex to efficiently render in OpenGL’s Stored Mode. It will instead be rendered in OpenGL’s Immediate Mode (can be less quick to update, but uses less memory)

```cpp
GetMaterial
```

By default, the logical volumes you create will get their material from the material parameter you specified for this component. But you can use GetMaterial to obtain any other named material.
First line of the cc file must be of the form:

```cpp
// Particle Source for MyParticleSource1
or
// Particle Generator for MyParticleGenerator1
```

Your particle source defines the initial particles that are then transported by the simulation. Because Geant4’s multithreaded capability keeps part of this functionality in the master thread and other parts in the worker threads, you actually create two separate classes to create a particle source.

For the part of the source that controls overall behavior (usually just setting the number of histories, but optionally also things like reading in some kind of phase space file), you write a class derived from TsSource. TOPAS instantiates this in the Geant4 Master thread. If you really just need this class to set the number of histories, you may just use our existing TsSource (that is, you don’t have to write your own class at all for this part).

For the part of the source that generates the individual events (setting the primary particle positions and momenta), you write a class derived from TsVGenerator. TOPAS instantiates this in the Geant4 worker thread.

In both cases, parameter lookups should be done in ResolveParameters. Call ResolveParameters directly from your constructor, and then you can also rely on TOPAS to re-call this method any time one of this particle source’s parameters is changed.

TOPAS will call your GeneratePrimaries method once per history. You should always start this method with this test:

```cpp
if (CurrentSourceHasGeneratedEnough()) return;
```

This allows your source to properly coexist with other sources that may have other numbers of histories.

The body of your GeneratePrimaries method should create and fill some number of TsPrimaryParticles (a single history may contain zero, one or more primary particles).

The TsPrimaryParticle structure is defined in the header file TsVParticleSource.hh. For each TsPrimaryParticle that you define, call GenerateOnePrimary.

Once you have finished creating all of the TsPrimaryParticles for this history, call AddPrimariesToEvent.
CHAPTER 31

Custom Physics Lists

First line of the cc file must be of the form:

```cpp
// Physics List for MyPhysicsList1
```

You can supply your own physics list. Note however that this option is not recommended unless you have significant Geant4 expertise. Even most long-time Geant4 users get into difficulty writing their own physics lists. Wherever possible, you should try to use one of the Reference physics list or the Modular physics list.

The example physics list provided in topas/extensions/TsPhysicsList1 includes a pointer to the TOPAS parameter manager as its argument. This is not required, but allows you to use TOPAS parameters to adjust options within your physics list.
Custom Scorers

First line of the cc file must be of the form:

```cpp
// Scorer for MyScorer1
```

Your custom scorer can either accumulate binned data (like our built-in dose scorer), or n-tuple data (like our built-in phase space scorer).

- For binned scorers, your scorer should inherit from `TsVBinnedScorer`.
- For n-tuple scorers, your scorer should inherit from `TsVNtupleScorer`.

At a minimum, your scorer should provide a constructor, a destructor and a `ProcessHits` method. The base class will take care of all the details of filtering, accumulating and outputting results.

For binned scorers, your scorer’s constructor must contain a call to:

```cpp
SetUnits
```

For n-tuple scorers, your scorer’s constructor defines each column and its data type by calls to:

```cpp
RegisterColumnD
RegisterColumnF
RegisterColumnI
RegisterColumnI8
RegisterColumnB
RegisterColumnS
```

`RegisterColumnD` and `RegisterColumnF` also take a unit string.

If your scorer is a Surface Scorer, the constructor must also contain the line:

```cpp
SetSurfaceScorer();
```

Otherwise, your scorer is assumed to be a Volume Scorer.

The scorer’s `ProcessHits` method must be written carefully to avoid slowing down the simulation since this method is called for every hit in the scoring component. Slow operations such as string comparisons should be avoided here.
Try to write your code so that you perform these sorts of slow operations only during construction, save values and pointers in class variables and then use these pre-calculated values in the ProcessHits method. Once you have your value computed:

- For binned scorers, accumulate data by calling AccumulateHit
- For n-tuple scorers, accumulate data by calling fNtuple->Fill

If you want to take more complete control of the scoring process, you can provide optional methods:

```cpp
// called after the last hit of a given track
void UserHookForEndOfTrack()
// called after the last hit of all tracks resulting from a given particle incident
// on the scoring component
void UserHookForEndOfIncidentParticle()
// called at the end of the event
void UserHookForEndOfEvent()
// called at the end of the run
void UserHookForEndOfRun()
```

Between the ProcessHits method and these other four methods, you have complete control over how you will accumulate and handle your scored values. Accumulate values in your own data structures that you provide in your scorer’s header file or in other classes that your scorer calls. Manipulate and output these values as you wish. It is all up to you. You can still choose to fill the fEvtMap just like a regular scorer, in which case TOPAS will accumulate and output those values, or you can fill nothing into that fEvtMap, in which case TOPAS will not take any further action for this scorer.

Some helper functions you may want to use from the TsParameterManager:

```cpp
// Activates creation of the TsTrackInformation object
SetNeedsTrackingAction
// Activates creation of the extra part of the TsTrackInformation object that contains information on what volumes were traversed
SetNeedsSteppingAction
```

Some helper functions you may want to use from the TsVScorer:

```cpp
// Get pointer to a material
GetMaterial
// Tell whether a given material is used in the geometry
UsedMaterial
// Get the voxel index from hits in divided or parameterized components
GetIndex
// Get the current TOPAS time (for the time of flight, use fTimeOfFlight)
GetTime
GetRunID
GetEventID
GetRandomNumberStatusForThisEvent
// Disable the automatic creation and filling of output, leaving this work entirely to your scorer
SuppressStandardOutputHandling
```

For divided components, the combined index one finds in scorers is formed from three bin indices (x,y,z or r, phi, z or r, phi, theta for TsBox, TsCylinder and TsSphere respectively). A helper function is now provided to return the individual bin indices given the combined index:

```cpp
GetBin(index, iBin) // where iBin is 0, 1 or 2
```
First line of the cc file must be of the form:

```cpp
// Filter for OnlyIncludeParticlesOfTwiceAtomicNumber,
// OnlyIncludeParticlesNotOfTwiceAtomicNumber
```

Note that a single filter can be used for more than one filter condition, hence comma separated list.

Filters must be written carefully to avoid slowing down the simulation. The filter’s Accept method is called for every hit in the scoring component. Slow operations such as string comparisons should be avoided during this method. Try to write your code so that you perform these sorts of slow operations only during the constructor, ResolveParameters method or CacheGeometryPointers method, save values and pointers in class variables and then use these pre-calculated values in the ProcessHits method.

Parameter lookups should be done in ResolveParameters. Call ResolveParameters directly from your constructor, and then you can also rely on TOPAS to re-call this method any time one of this filter’s parameters is changed.

4D behaviors may require TOPAS to destroy and rebuild components during the simulation. Accordingly, you can not rely on the pointer to a given component remaining the same throughout the simulation. Any lookup of a component pointer should be done in the filter’s CacheGeometryPointers method. TOPAS will re-call this method any time relevant components are rebuilt.

Some helper functions you may want to use from the TsParameterManager:

```cpp
// Activates creation of the TsTrackInformation object
SetNeedsTrackingAction
// Activates creation of the extra part of the TsTrackInformation object that ...
// contains information on what volumes were traversed
SetNeedsSteppingAction
```

Some helper functions you may want to use from the TsVFilter:

```cpp
// Get pointer to a material
GetMaterial
// Get pointer to a named physics volume
GetPhysicalVolume
// Get pointer to a named component
```
GetComponent
// Get pointers to all children of a named component
GetChildComponentsOf
While an ElectroMagnetic Field can have just an electric field, just a magnetic field, or a combination of the two, Geant4’s architecture does not provide any base class for a purely electric field, but instead provides one base class for purely Magnetic fields and another for combined ElectroMagnetic fields. To allow you to use all features of both Geant4 classes, TOPAS emulates this curious aspect of Geant4’s design.

If you want to create a purely magnetic field, the first line of the cc file should be of the form:

```c++
// Magnetic Field for MyField1
```

If you want to create a purely electric field, or a combined electromagnetic field, the first line of the cc file should be of the form:

```c++
// ElectroMagnetic Field for MyField1
```

and then if what you really wanted was just an electric field, you implement the magnetic field strength as just zero.

Geant4 will call your GetFieldValue every time it needs to query the field. For reasons that are not clear to this author, Geant4 will sometimes query your field for points outside of your intended geometry component, so make sure to return at least some value (at least a zero) for every possible point.

Parameter lookups should be done in ResolveParameters. Call ResolveParameters directly from your constructor, and then you can also rely on TOPAS to re-call this method any time one of this field class’s parameters is changed.
Custom Imaging to Material Conversion

First line of the cc file must be of the form:

```
// Imaging to Material Converter for MyImagingToMaterialConverter1
```

You can supply your own class to assign imaging values to materials.

To use your Imaging to Material converter, reference its name in the parameter:

```
G::Ge/Patient/ImagingtoMaterialConverter = "MyImagingToMaterialConverter1"
```

The number of image files read by Topas is determined by the parameter:

```
G::Ge/Patient/NumberOfEnergies = 1 # defaults to 1
```

If this value is just 1, Topas will look for imaging files directly in your DicomDirectory. If this value is larger, Topas
will expect your DicomDirectory to contain numbered subdirectories:

```
YourDicomDirectory/1
YourDicomDirectory/2
etc.
```

The allowed modalities of the imaging files is determined by the parameter:

```
Gv::Ge/Patient/DicomModalityTags = 1 "CT" # defaults to just CT
```

Other modality tags are, for example, MR for Magnetic Resonance and US for Ultrasound. A complete list can be
found here.

You apply whatever algorithm you like in your class’s AssignMaterial method.

This will be called once for each voxel. TOPAS will pass you a vector of imagingValues, with each value representing
this voxel’s value from one of your image values:

```
std::vector< signed short > imagerValues
```
For example, if you are doing Dual Energy CT, you will get two values in this vector, the HU values from each of the two CT files.

The materials you use can be defined in your parameter file, in your ImagingToMaterial class’s constructor or in your ImagingToMaterial class’s AssignMaterial method. Either way, by the time you are finished assigning all of your materials, you will have built up a vector of pointers to materials in fMaterialList and, for each voxel, your AssignMaterials method will have returned an appropriate index into this vector.

AssignMaterials is also passed a timeSliceIndex, which is useful if your imaging is time-dependent (that is, 4D imaging). In this case you can use the timeSliceIndex however you wish in your AssignMaterials algorithm.

To avoid spending CPU time on repeated parameter lookups, it is best to do them in ResolveParameters. Call ResolveParameters directly from your constructor, and then you can also rely on TOPAS to re-call this method any time one of this class’s parameters are changed.
Six additional hooks are provided in TsExtensionManager for you to attach your own code.

- BeginSession
- BeginRun
- BeginHistory
- EndHistory
- EndRun
- EndSession

First line of the cc file must be of the form:

```c++
// BeginSession for TOPAS
// BeginRun for TOPAS
// BeginHistory for TOPAS
// EndHistory for TOPAS
// EndRun for TOPAS
// EndSession for TOPAS
```

There are no particular constraints on what you can do in these methods. They are provided simply to give you more flexibility in the design of your extensions.