Quick Guide to the MCTAL API

(MCTAL ⟶ ROOT)

Nicolò Borghi

This document is a short quick guide to the usage of the MCTAL API for converting MCTAL files produced with MCNPX into the binary format of ROOT files.

1 Software requirements

In order to use this tool, the following mandatory packages are needed:

1. Python $\geq$ 2.7.3
2. ROOT Data Analysis Framework (Download from CERN)[1] compiled with Python support
3. ndiff (required only for conversion test) (Download)[2]

2 Converting MCTAL files

Assuming that the MCTAL output is stored in the `mctal` file, the conversion script command-line syntax is very simple:

```
./nbmctal2root.py mctal
```

This line will create a file with the same name of the input MCTAL and with a trailing `.root`. In this case, the file name will be `mctal.root` and it will be saved in the same directory of the input MCTAL.

```
ROOT file saved to: mctal.root
```

It is also possible to provide an absolute path to the `nbmctal2root.py` script and the output line will change accordingly.

A different file name for the converted file can be requested, by adding the desired file name after the input MCTAL (also with the desired path):

```
./nbmctal2root.py mctal examplename.root
```

A verbose output of the conversion is available with the option `-v` (it must appear before any other arguments):

Then, the output will resemble the following:

[1] http://root.cern.ch/drupal/content/downloading-root

[2] http://www.math.utah.edu/~beebe/software/ndiff/

```
./nbmctal2root.py -v  mctal

[Parsing file: mctal...]
 Parsing tally:    15

[Converting...]
 Tally    15 saved

ROOT file saved to: mctal.root
```

The ROOT file will contain a **THnSparseF** object named after the tally number (one **THnSparseF** for each tally) with a leading `f`. Hence, tally `15` will be saved as a **THnSparseF** called `f15`. Each **THnSparseF** has eight dimensions, according to the binning capabilities of MCNPX:

```
F Surface/cell/detector F card bin
└── D Total/direct vs flagged/unflagged bin
    └── U User bin
        └── S Segment or radiography s-axis bin
            └── M Multiplier bin
                └── C Cosine or radiography t-axis bin
                    └── E Energy bin
                        └── T Time bin
```

In the case of mesh tallies, everything holds, but there will be three more dimensions:

```
└── I First lattice/mesh index
    └── J Second lattice/mesh index
        └── K Third lattice/mesh index
```

3 Testing the reading and the conversion

In order to check if the conversion from MCTAL to ROOT is correct, a set of test routines is available. With the script `nbmctaltest.py` it is possible to check whether the reading of the MCTAL file into a Python object was successful or not:

```
nbmctaltest.py mctal
```

It is possible to add a `-v` option before the MCTAL file name in order to see detailed output. If no output is printed, then the test is successful. The script returns the exit code 0 on success and 1 on failure.

The test of the conversion to the ROOT format is implemented in the `conversiontest` bash script. This script is divided in two steps:

1. `nbmctal2root.py` is called for conversion

2. `nbroottest.py` is called for test

The command-line syntax is the following (the -v option is also available and must be set before the MCTAL file name):

```
conversiontest mctal
```

An exit code is returned: 0 for success, 1 for failure.