



Some Additional *art* Facilities

Chris Green.
art/ LArSoft course.
August 6, 2015.



Fermi National Accelerator Laboratory
Office of Science / U.S. Department of Energy
Managed by Fermi Research Alliance, LLC

*Command-Line Options to the
art Family of Programs*

Command-Line Options — Introduction

- Most important: `-h` | `--help`.
- Most command-line options modify the configuration as presented initially by `--config`, which makes `--debug-config` and `--config-out` useful for understanding what they do.

Command-Line Options — Informational options

- **--debug-config** <output-cfg-file> — Output the post-processed configuration to the specified file, then exit.
- **--config-out** <output-cfg-file> — Output the post-processed configuration to the specified file, then continue according to said configuration.
- **--[no]tracer** — [De-]activate the *Tracer* facility; equivalent to
`services.scheduler.wantTracer :true | false`
in the configuration.
- **--[no]memcheck** — [De-]activate the **SimpleMemoryCheck** service. This is deprecated: the newer **MemoryTracker** is more versatile.
- **--print-available-(modules|services)** — print available modules or services, and their details.
- **--(module|service)-description** (repeatable) — print a description for the specified module or service by type.

Command-Line Options – Basic options

- **-c|--config** **<config-file>** — specify the top-level **FHiCL** file for the job.
- **-s|--source** **<input-file>** (repeatable) — specify input files, processed in given order. If **<config-file>** does not specify the type of input, it is assumed to be **RootInput**. Input files may also be specified as trailing non-option arguments.
- **-o|--output** **<output-file>** — specify a single output file. Assumes at most one output stream as specified by **<config-file>**; if unconfigured, the output module is assumed to be **RootOutput**.
- **-T|--TFileName** **<hist-file>**
— specify the output file for the **TFileService**. Equivalent to `services.TFileService :{fileName :<hist-file> }` in the **FHiCL** configuration.

Command-Line Options – Other options

- **--process-name** `<process-name>` — equivalent to `process_name :<process-name>` in the **FHiCL** configuration.
- **-S|--source-list** `<list-file>` — `<list-file>` contains a list of input files to use (*c.f.* **-s**).
- **--tmpdir** `<dir>` — in-progress output files will be written here before being moved to their final locations on close.
- **--sam-*** `<...>` — specify various configuration items for **SAM** input and metadata for **SAM** upload of output files.
- **--default-exceptions**, **--rethrow-**(`default|all`) — gross control of framework exception handling.
- **--nskip** `<#>`, **-n|--nevts** `<#>`, **-e|--estart** `<#>` — control which and how many events to process.

Some Pointers for Use of Command-Line Options.

- Use `--rethrow-all` wherever possible, especially in production.
- Follow your experiment's guidelines, but in general try to put as much configuration as possible into the `FHiCL` file(s), reserving command-line options for items that change per invocation (especially for production), or for diagnostic changes.

*Auxiliary Programs:
config_dumper and
sam_metadata_dumper*

config_dumper

Access to configuration information stored in **art ROOT**-format files.

- **-h|--help** — help.
- **-M|--modules** | **-S|--services** | **-P|--processes** — Print configurations by module, service or process.
- **-f|--filter** <arg> (repeatable) — filter results by module label, service type or process name, as appropriate.
- **-s|--source** <file> (repeatable) — specify **art ROOT**-format files to read.

sam_metadata_dumper

Access to **SAM** metadata stored in **art ROOT**-format files.

- **-h|--help** — help.
- **-H|--hr|--human-readable** — produce human-readable output rather than the default **JSON**.
- **-s|--source <file>** (repeatable) — specify **art ROOT**-format files to read.

*Some Useful **art**-Provided
Services*

art-Provided Services

art provides several services for general use. Of particular interest to the physicist user are **TimeTracker** and **MemoryTracker**. Note that older, simpler services exist: **Timing** and **SimpleMemoryCheck** which can provide some information as part of the job's output. However, the newer services are more versatile and provide information additionally in the form of an **SQLite** database which can be interrogated interactively or even used in systems such as **R** or **ROOT** for graphing and analysis. The command-line executable **sqlite3** is also available as part of the **art** suite.

art-Provided Services

Common features of **TimeTracker** and **MemoryTracker**:

- Configurable end-of-job summary (via **messagefacility** message).
- Output to an **SQLite** file.

TimeTracker

Configuration:

`services.TimeTracker` : {<parameters> }, where <parameters> can be:

- `printSummary` : `true` | `false` (default `true`).
- `dbOutput` : { `filename` : <filename> } (default `none`).
- `dbOutput` : { `overwrite` : `true` | `false` } (default `false`).

TimeTracker printout (sec)	Min	Avg	Max	Median	RMS	nEvs
Full event	3.5883e-05	5.90957e-05	0.0086342	4.0041e-05	0.000299869	1000
pl:prod:TestTimeTrackerProducer	8.05e-07	1.07218e-06	1.4539e-05	9.3e-07	1.18794e-06	1000
pl:filt:TestTimeTrackerFilter	6.19e-07	7.23938e-07	1.3557e-05	6.82e-07	5.85689e-07	1000
pl:TriggerResults:TriggerResultInserter	1.017e-05	1.21662e-05	0.000136089	1.10565e-05	5.17438e-06	1000
end_path:mod1:TestTimeTrackerAnalyzer	3.938e-06	4.83591e-06	0.000104646	4.339e-06	3.7656e-06	1000
end_path:mod2:TestTimeTrackerAnalyzer	4.25e-07	5.35696e-07	1.3921e-05	4.71e-07	8.4755e-07	1000

TimeTracker

Example SQLite session:

```
$ sqlite3 timeTracker.db
SQLite version 3.8.8.2 2015-01-30 14:30:45
Enter ".help" for usage hints.
sqlite> .tables
TimeEvent  TimeModule  TimeReport
sqlite> .schema
CREATE TABLE TimeReport ( ReportType text, Min numeric, Mean numeric,
                           Max numeric, Median numeric, RMS numeric,
                           nEvts integer );
CREATE TABLE TimeEvent ( Run integer, Subrun integer, Event integer,
                           Time numeric );
CREATE TABLE TimeModule ( Run integer, Subrun integer, Event integer,
                           PathModuleId text, Time numeric );
sqlite> .q
```

MemoryTracker

Configuration:

`services.MemoryTracker` : {<parameters> }, where <parameters> can be:

- `printSummary` : [<spec>], where <spec> can be "*" or any combination of, "general", "event", and/or "module" (default ["*"]).
- `filename` : <filename> (default none).
- `numToSkip` : <n> — number of initial events to exclude from memory (default 1).
- `includeMallocInfo` : `true` | `false` — (default `false`).

N.B. MemoryTracker is dummied on OS X: your experiment should ask for this as a feature if you need it!

MemoryTracker

MemoryTracker example output (per-event summary):

MemoryTracker Per-event SUMMARY

(Numbers in '[...]' correspond to the event nos. in the general summary.)

Events increasing Vsize (Mbytes)	Vsize	Δ Vsize	RSS	Δ RSS
[3] run: 1 subRun: 0 event: 3	346.727	1.148	48.633	1.148
[4] run: 1 subRun: 0 event: 4	347.875	1.148	49.781	1.148
[5] run: 1 subRun: 0 event: 5	349.023	1.148	50.930	1.148
[6] run: 1 subRun: 0 event: 6	350.172	1.148	52.078	1.148
[7] run: 1 subRun: 0 event: 7	351.320	1.148	53.227	1.148

Events with large Vsize (Mbytes)	Vsize	Δ Vsize	RSS	Δ RSS
[10] run: 1 subRun: 0 event: 10	354.766	1.148	56.676	1.148
[9] run: 1 subRun: 0 event: 9	353.617	1.148	55.523	1.148
[8] run: 1 subRun: 0 event: 8	352.469	1.148	54.375	1.148
[7] run: 1 subRun: 0 event: 7	351.320	1.148	53.227	1.148
[6] run: 1 subRun: 0 event: 6	350.172	1.148	52.078	1.148

MemoryTracker

Example SQLite session:

```
$ sqlite3 memoryTracker.db
SQLite version 3.8.8.2 2015-01-30 14:30:45
Enter ".help" for usage hints.
sqlite> .tables
EventInfo          ModuleInfo          Summary
EventMallocInfo    ModuleMallocInfo
sqlite> .schema
CREATE TABLE Summary ( ProcessStep text, ModuleId text, DeltaVsize numeric,
                        DeltaRSS numeric );
CREATE TABLE EventInfo ( Run integer, Subrun integer, Event integer,
                          Vsize numeric, DeltaVsize numeric, RSS numeric,
                          DeltaRSS numeric );
CREATE TABLE ModuleInfo ( Run integer, Subrun integer, Event integer,
                           PathModuleId text, Vsize numeric,
                           DeltaVsize numeric, RSS numeric,
                           DeltaRSS numeric );
CREATE TABLE EventMallocInfo ( EvtRowId integer, arena integer,
                                 ordblks integer, keepcost integer,
                                 hblkhd integer, hblks integer,
                                 uordblks integer, fordblks integer );
CREATE TABLE ModuleMallocInfo ( ModRowId integer, arena integer,
                                  ordblks integer, keepcost integer,
                                  hblkhd integer, hblks integer,
                                  uordblks integer, fordblks integer );
sqlite> .q
```