MCFast Commands

Abstract

This document contains a description of MCFast commands and command files.

MCFast v2_5_1 and v2_6

1 Running MCFast

The fast Monte Carlo “MCFast”is executed by typing

mcfast [-f command_file] [-l lpt_file] [-hb Hbook_file]

where

command_file = name of file containing job directives
lpt_file = name of ASCII output file
Hbook_file = name of output Hbook file

Note: if the command file is not specified, mcfast accepts directives from the terminal, terminated by a $\wedge$D (end of file). This is very convenient when running multiple jobs in batch.

You can get online help on running mcfast by typing

mcfast -h

with no other arguments.

During execution, mcfast checks the user’s terminal every event to see if anything has been typed. This feature can be disabled for batch jobs by entering the line

batch ON

in the command file. Commands can be abbreviated to the minimum necessary for uniqueness. Type “help” during execution to get a list of commands. Note that if you type “user”, the routine usr_command is called. You can provide this routine to print out any information you want while the program is running.

List of commands that MCFast recognizes during execution(can be abbreviated):

continue Continue after a pause

exit/quit/stop Stop reading data and finish

help Print this message

hcdir $< dir >$ Change Hbook directory

hldir List contents of current Hbook directory

htype $< \# >$ Type Hbook histogram in current directory

next Analyze next event and pause

pause Pause and wait for further commands

skip $<$ n $>$ Read in but do not analyze the next n events

status Print a brief status message to terminal

user Call "usr_command" (supplied by user)

2 MCFast Command Files

The ommand file format is as follows:

input file format:
keyword *value*
or keyword *value*1 *value*2

where keyword is one of the variables that can be set.

A whole series of reals, integers and strings can be read in. See the routine *cmd_getitems* for more details.

A sample command file for the C0 detector follows:

```
! C0 command file – no transverse shower development, no xdr output
!
! use a ! to start a comment
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
max_ev 20000 ! Max number of events to analyze
max_print 2 ! max # events to print in lpt file
file_geo c0.db ! Geometry file
ranseed 0 ! Random seeds– if 0 then use time of day
file_type 1 ! file_type 1 is the default from evgen
file_in @psiks.fil ! @filename is the file containing a list of data files

!Example of simple trigger control
!
trigger
l1 define 0 track_trigger_1 use
l1 term track require num 1 pt 0.25 eta 1.5 5.0

l1 define 1 track_trigger_2 use
l1 term track require num 1 pt 0.25 eta -5.0 -1.5
```

```
l1 define 8 muon_trig_1 use
l1 term muon require num 1 pt 0.25 eta 1.5 5.0

l1 define 9 muon_trig_2 use
l1 term muon require num 1 pt 0.25 eta -5.0 -1.5
end trigger


! program control
! use TRUE, .TRUE., T, ON, YES, Y
! or FALSE, .FALSE., F, OFF, NO, N


control_shower
shw_trans OFF ! DO NOT generate shw.trans.prof. as shw propagates
e_thresh 0.05 ! % of remaining energy to stop tracing shw. – default
e_vis_min 0.002 ! minimum energy to read out in CAL cell – default
max_hit_step 25 ! max # of hits to generate per step – default
end control_shower


! The following 3 quantities are set True by default in MCFast
make_decays T ! do decays in flight
make_pair_convert T ! do pair conversions
make_hits T ! do generate hits
```

3 MCFast Command Definitions

The following terms are recognized by MCFast and can be entered in the command file.

max_event – Maximum number of event to be processed

max_print – Maximum number of event to be printed in lpt file

batch – set ON for batch jobs; default = OFF

file_in @*filename* – file containing list of signal files

background_in @*filename* – file containing list of background

file_out *filename* – output events in OLD STDHEP format– obsolete

file_type 1 – input signal data file format; default = 1

background_type – input background data file format; default = 1

ranseed – enter 2 seeds for random number generator (RANECU); default = 0.0 0.0 (if first seed is zero then MCFast uses time of day for seeds)

int_crossing – # of interactions per crossing

generate_internal – initiate particle gun

fcon $< \# >$ *value* – User defined floating point constant (500 allowed)

icon $< \# >$ *value* – User defined integer constant (500 allowed)

file_geometry *filename.db* – geometry file name

luminosity – default = 10^{32}

runtime – default = 10^{7}

trigger – initiate trigger definitions

dbg_trace – include debug statements in MCFast; default = OFF – see section on DEBUG commands below

dbg_hist – include debug histograms in MCFast; default = OFF – see section on DEBUG commands below

qq_user_file – change QQ user file for decays in flight; default = $MCFAST_DIR/examples/mcfast/stable.dec

make_decays – control flag for decays in flight; default = ON

make_pair_convert – control flag for pair conversions; default = ON

make_hits – control flag for hit generation; default = ON

use_mult_scat – contral flag for multiple scattering in trace (v2_6); default = OFF

use_energy_loss – contral flag for dedx in trace (v2_6) default = OFF

control_shower – initiate shower control parameters

step_max_distance – maximum distance between trace points (for graphics) default = 1.e10

step_max_angle – maximum angle between trace points (for graphics)1.e10

step_max_dedx – maximum step for dedx (v2_6) default = 1.e10

step_max_mscat – maximum step for scattering (v2_6) default = 1.e10

trk_max_turns – maximum number of turns per step – default = 0.5

trk_min_kinetic – minimum kinetic energy for track – default = 1.e6 MeV

mcfio_out – mcfio output file name

geomview_out – geomview output file name – this is obsolete.

4 Particle Gun Command Definitions

multiplicity $< \# >$– average multiplicity; default = 1

fix_multiplicity ON – default; if OFF then use Poisson distribution

slope *slope* – default = 0.; if $slope \neq 0$ then $p_t = ptmin - log(1 - random\# \times (1 - exp(-slope \times (ptmax - ptmin))))/slope$

type_info *hep_id probability mass* – particle information; default = [211 1.0 0.13956]

ptmin *ptmin* – minimum p_t ; default = 1.

ptmax *ptmax* – maximum p_t ; default = 1.

etamin *etamin* – minimum eta; default = 0.

etamax *etamax* – maximum eta; default = 0.

phimin *phimin* – minimum phi angle; default = 0.

phimax *phimax* – maximum phi angle; default = 2π.

vertex *x y z* – primary vertex position; default = [0.0 0.0 0.0]

dvertex *dx dy dz* –sigma of primary vertex position; default = [0.0 0.0 0.0]

5 Debug Flags

Used with command **dbg_trace** and **dbg_hist**:
dbg_trace muon ON ; Turn on debug output for muon code
dbg_hist muon ON ; Turn on debug histograms in muon code

The default is OFF for all subsystems.

BEWARE! Some of these switches turn on lots of output. Many of them do nothing.

general ON sets dbg_trace(jdbg_general) .true. or dbg_hist(jdbg_general)

track ON sets dbg_trace(jdbg_track) .true. or dbg_hist(jdbg_track)

shower ON sets dbg_trace(jdbg_shower) .true. or dbg_hist(jdbg_shower)

gamma ON sets dbg_trace(jdbg_gamma) .true. or dbg_hist(jdbg_gamma)

emcal ON sets dbg_trace(jdbg_emcal) .true. or dbg_hist(jdbg_emcal)

muon ON sets dbg_trace(jdbg_muon) .true. or dbg_hist(jdbg_muon)

trig ON sets dbg_trace(jdbg_trig) .true. or dbg_hist(jdbg_trig)

geom ON sets dbg_trace(jdbg_geom) .true. or dbg_hist(jdbg_geom)

hits ON sets dbg_trace(jdbg_hits) .true. or dbg_hist(jdbg_hits)

6 Shower Control Command Definitions

shw_trans – Default value is ON. The transverse profile of showers are generated by default. In order to speed up the program, it is possible to create calorimeter hits only on the exact path of the shower core.

e_thresh – The default value is 0.05. With the current showering algorithm, we stop tracing a shower if the remaining energy of the shower is 5% or less of the incident energy.

max_hit_step – The default value is 25. This is the maximum number of hits with status SHOWERING that can be produced over each step as a shower propagates. Correlated with **e_vis_min** (nhits = min((edep/e_vis_min),max_hit_step)); max_hit_step = 0 is equivalent to shw_trans OFF. The user may change this parameter in order to speed up the code, but the precision of the shower transverse profile generation may be affected.

e_vis_min – The default value is 0.002 GeV. This is the minimum energy that can be recorded in the calorimeter cell (currently it is set once for all calorimeters.)

7 Trigger Command Definitions

Three types of triggers can be defined at a single level in the MCFAST framework. The trigger types are TRACKING, MUON, and EM cluster. These objects work only from the generator information.

To define triggers, several pieces of information must be specified.

1. The level of the trigger (only L1 is active in this release)
2. A key word corresponding to the function required

   define – sets up a specific trigger

   term – sets up the conditions corresponding to that specific trigger

   For "define" programming:

   a) bit number– currently 0-31

   b) Name for the trigger

   c) Flag to determine whether or not to cut on the trigger condition.

Examples:

l1 define 2 track1_trigger use

l1 –Trigger level

define –Key word

2 –specific bit number (0-31)

track1_trigger –trigger name–should be <32 characters

use – cut on trigger condition, if left off, accounting for trigger will still appear in summary, but trigger p/f will not be used.

Note that at this point, the trigger doesn't know anything about what is required. Ideally, the name should have some meaning to the user, but what is used in the name has no effect on the trigger requirements.

For "term" programming:

a) A key word corresponding to the type of trigger requested

muon

track

calem

b) require or veto on this condition

c) number of objects

d) pt of object(s)

e) eta of object(s) – For a symmetric geometry, be sure to specify - eta to + eta.

f) phi of object(s) – if not used, a 2π default is taken.

Example:

l1 term track require num 1 pt 1.5 eta -10. 10.

l1 – trigger level

term – key word

track – object type (track, muon or calem)

require – accept if the condition is true, use **veto** to accept if the condition is false

num 1 – multiplicity

pt 1.5 – pt threshold

eta -10. 10. – eta range

This supplies the requirements to the trigger. There can be up to 256 terms associated with each specific trigger. Examples of complete triggers are:

l1 define 0 test
l1 term track require num 1 pt 0 eta -10. 10.
just count the events with any tracks at all

l1 define 1 track1_trigger use
l1 term track require num 1 pt 1.5 eta 0. 10.

l1 define 2 track2_trigger use
l1 term track require num 1 pt 3 eta -10. 10.
l1 term track require num 2 pt 1.5 eta -10. 10.

l1 define 3 muon_trig use
l1 term muon require num 1 pt 2.0 eta 1. 4.
l1 term track require num 1 pt 2.0 eta 1. 4.
note, the muon and track are not "matched"

Output: At the end of the mcfast.lpt file, trigger summary is printed out showing a truncated version of the term definitions, the number of events tried and passed for each term and each trigger. Table 1 shows the output from a typical job with 2 track triggers, 1 muon trigger and 4 calorimetry triggers defined.

Table 1: Trigger Programming Summary: level 1 term definitions

TERM	TERM NAME	events passed /	tried
1	l1 term track require num 1 pt 1.5 eta 0	347/	500
2	l1 term track require num 1 pt 1.5 eta -	485/	500
33	l1 term muon require num 1 pt 2.0 eta 0.	168/	500
65	l1 term calem require num 1 pt 0.0 eta -	335/	500
66	l1 term calem require num 1 pt 0.0 eta 0	221/	500
67	l1 term calem require num 1 pt 0.1 eta -	49/	500
68	l1 term calem require num 1 pt 0.1 eta 0	23/	500

LEVEL 1 specific triggers
Trigger 0 TRACK_TRIGGER required 347/ 500
terms used: 1
Trigger 1 TRACK1_TRIGGER required 485/ 500
terms used: 2