cmsDriver.py

Introduction

cmsDriver is a tool to create production-solid configuration files from minimal command line options. For example the command

```
cmsDriver.py SingleMuPt1_cfi.py -s GEN,FASTSIM --conditions=MC_3XY_V10::All --eventcontent=RECOSI
```

would create a configuration file that takes the single muon settings from `Configuration/Generators/SingleMuPt1_cfi.py` and then runs the generator specific parts of CMSSW followed by fast simulation. During this process, it will use the global tag `MC_3XY_V10::All` for applying alignment and calibration constants. The output of this job will contain RECOSIM content and will have 10 events.

Standard Steps for full simulation and real data

Building blocks of the created configurations are the standard processing steps:

- GEN : the generator plus the creation of GenParticles and GenJets
- SIM : Geant4 simulation of the detector (energy deposits in the detector volumes)
- DIGI : simulation of detector signal response to the energy deposits
- L1: simulation of the L1 trigger
- DIGI2RAW : data format conversion of the digi signals into the RAW format that will be provided in the online system
- HLT : high level trigger

Usually all the above steps are executed in one single job. Remaining building blocks are:

- RAW2DIGI : data format conversion of the RAW format into digi signals
- RECO : full event reconstruction
- ALCA : production of alignment and calibration streams
- DQM : code run for DQM
- VALIDATION : code run for validation

The above list is usually referred to as 'step2'.

Standard steps for fast simulation

For fast simulation, things are slightly different. As the fast simulation combines a lot of things, there are only two standard steps:

- GEN
- FASTSIM

Command line options

A summary of the cmsDriver.py script s options with a detailed message about each one can be visualized by getting the help:

```
cmsDriver.py --help
```

The options list is divided into two sections according to the user s level of knowledge: Options and Expert Options. The latter is addressed to expert users.

Standard Options

`-s STEP, --step=STEP`: this option is useful to indicate what kind of steps the user wants to run and in which order. The possible values are: GEN,SIM,DIGI,L1,DIGI2RAW,HLT,RAW2DIGI,RECO,POSTRECO,DQM,ALCA,VALIDATION,HARVESTING or ALL. Look at Standard Steps for full simulation and real data for further informations. An example of usage is: `-s GEN,SIM,DIGI,L1,DIGI2RAW,HLT,RAW2DIGI,L1Reco`. One or more filters to run on generator level can also be specified: `-s GEN:ProductionFilterSequence,SIM,DIGI,L1,DIGI2RAW,HLT,RAW2DIGI,L1Reco`.

For the `HLT` step, one can specify a HLT configuration after a colon (e.g. `HLT:GRun`). The files corresponding to the HLT configuration string are in HLTrigger/Configuration/python (files with `Famos` in their name are used for fast simulation).

`--conditions=CONDITIONS`: this option concerns the alignment and calibration conditions the user wants to apply for producing the dataset: `--conditions=MC_3XY_V10::All`. The default value is set to frontier conditions STARTUP_V4::ALL (FrontierConditions_CMS.GlobalTag,STARTUP_V4::All). For a list of all available tags refer to SWGuideFrontierConditions.

`--eventcontent=EVENTCONTENT`: the user can select what event content has to be written out in the output by making use of this option: `--eventcontent=RECOSIM`. Default values are FEVTDEBUG or FEVT for cosmics. The user can choose among those available in the files in Configuration/EventContent/python/ directory.

`--filein=FILEIN`: specifying this option the user can indicate the infile name. For instance, if in the previous step the processing of events has been run up to the HLT, in the next step the reconstruction can be run. In this case the user has to specify what was the output file of the previous generation as infile for the new configuration file: `--filein file:YOUR-PHYSICS-CHANNEL_CMS-ENERGY_cff_py__GEN_SIM_DIGI_L1_DIGI2RAW_HLT.root \` or in general the user has to specify this option if he want to produce a full config file running on a already existing dataset.

`--fileout=FILEOUT`: this option can be used to customize the name for the output file to specify in the configuration file created by cmsDriver.py.

`-n NUMBER, --number=NUMBER`: indicates the number of events to write out in the event content of the output file. The default is 1.

`--mc`: this option, if defined, imposes the processing of simulation. A default value is based on all other defined options.

`--data`: this option, if defined, imposes the processing of real data. A default value is based on all other defined options.

Caution: if neither `--mc` option nor `--data` one are specified it will be determined that the user is requiring for simulation.

`--no_exec`: this option implies to prepare the full python configuration file without execute the `cmsRun` at the end.

Expert Options

`--beamspot=BEAMSPOT`: the user can specify this option to indicate what beam spot to use choosing from Configuration/StandardSequences. Default depends on scenario.

Example: `--beamspot=Realistic8TeVCollision`

`--customise=CUSTOMISATION_FILE`: use this option to specify the file where the code needed to modify the process object is stored.

`--datatier=DATATIER`: specifies what is the data tier to use.

`--dirin=DIRIN`: indicates in which directory the infile is stored.

`--dirout=DIROUT`: indicates in which directory the outfile is stored.

`--filtername=FILTERNAME`: indicates what is the filter name to specify in output module.

`--geometry=GEOMETRY`: the argument of this option is the geometry to use among those available in Configuration/StandardSequences. The default value is Ideal.

`--magField=MAGFIELD`: this option refers to which is the magnetic field to use to produce the dataset. It has to be choosen among values available in Configuration/StandardSequences.

`--no_output`: specifies to not write anything to disk. This option is intended for benchmarking purposes.

`--oneoutput`: specifies to use only one output module.

`--prefix=PREFIX`: the user can apply this option to set a prefix to the `cmsRun` command.

`--relval=RELVAL`: indicates the total number of events and events per job to set. An example of usage is: `--relval 25000,250`. This indicates the total number of events required is 25000, divided in 250 events per job.

`--dump_python`: this option is useful to dump the config file in python and do a full expansion of imports.

`--dump_DSetName`: the user can call this option to ask for the name of the primary dataset.

`--pileup=PILEUP`: call this option to set the pileup configuration to use. The default value is `NoPileUp` . See Configuration/StandardSequences/python/Mixing.py for the known pileup scenarios.

`--python_filename=PYTHON_FILENAME`: the user can choose to move the name of the created config file to `PYTHON_FILENAME` by using this option.

`--secondfilein=SECONDFILEIN`: specifies the name of the secondary infile for the two-file solution. The default is set to no file.

`--writeraw`: indicates to write out a secondary file, in addition to the nominal output, with just raw.

`--processName=NAME`: this option sets the process name to `NAME`.

`--scenario=SCENARIO`: selects the specified scenario to override the standard settings. Available values are: [pp , cosmics , nocoll , 'HeavyIons'].

`--harvesting=HARVESTING`: specifies what is the harvesting to use in the dataset production. Possible values can be choosen from Configuration/StandardSequences. The default value is set to `AtRunEnd`.

Advice on using cmsDriver.

- It will fail unless you have done "scram b" in your top-level CMSSW_x_y_z/src/ directory.
- If you are using it to do the GEN step, try doing "python -i" on your python generator fragment before you run cmsDriver on it. This will check it has no python errors.
- If you are using it to do the GEN step, it will fail if the python generator fragment that you are running it on includes the symbol "." in its file name (not counting the final ".py" in its file name).

Examples

- cmsDriver.py
  Configuration/GenProduction/python/ThirteenTeV/LongLivedChi0ToNuLL_Template_TuneCUETP8M1_pyt
  --mc --eventcontent RAWSIM --customise
  SLHCUpgradeSimulations/Configuration/postLS1Customs.customisePostLS1,Configuration/DataProcessing/U
  --datatier GEN-SIM --conditions MCRUN2_71_V1::All --beamspot NominalCollision2015 -s
  GEN,SIM --magField 38T_PostLS1 -n 20 --no_exec

Links

- main code used by cmsDriver.py (location for older CMSSW releases)

-- BenediktHegner - 14-Oct-2009

-- AnnapaolaDeCosa - 15-Oct-2009

This topic: CMSPublic > SWGuideCmsDriver
Topic revision: r13 - 2020-07-14 - JoshuaHiltbrand