

Table of Contents

Re-running the L1 emulator on existing samples.....	1
Workflow 1 : Re-running the full L1 emulator from raw data.....	1
Customization python script.....	2
Run with a different configuration.....	2
Run with the L1 trigger configuration from a different global tag.....	2
Run with the same global tag, but overwrite some of the L1 trigger records.....	2
Overwrite the L1 menu with a menu from an SQL file.....	2
Overwrite the L1 menu with a menu from an an XML files.....	3
Customization of output commands.....	3
Print the L1 trigger report.....	3
Obsolete: L1 trigger configured via dummy producers.....	3
Workflow 2 : Re-running the L1 GT emulator from raw data.....	3
Running L1 trigger emulator or L1 GT emulator and HLT.....	3
Review status.....	3

Re-running the L1 emulator on existing samples

There are two frequent use cases to re-run (parts of) the L1 trigger emulator:

- re-run the full L1 trigger emulator on raw data, with a different L1 trigger configuration
- re-run only the L1 Global Trigger emulator on digis, with a different L1 menu or/and with different prescale factors or/and with different trigger masks.

Starting from releases from the cycle 31X, the L1 trigger configuration is managed via the global tags, with content specific for each global tag.

Workflow 1 : Re-running the full L1 emulator from raw data

This allow the user to re-run L1 with the same or a different L1 configuration than that used to produce the original sample.

An example can be found at [L1Trigger/Configuration/test/L1EmulatorFromRaw.sh](#).

The script generates using cmsDriver a python file, which can then be used with cmsRun to produce the desired sample. All options of the cmsDriver can be used, users can modify the options in the script according to their needs.

The steps run are "RAW2DIGI,L1", with the customization needed to run the L1 emulator done in [customise_l1EmulatorFromRaw.py](#).

The job runs RawToDigi for calo and muon subdetectors, followed by the trigger primitive and central L1 trigger emulators. The only required input collection is the FEDRawDataCollection.

The following recipe can be used (tested on lxplus)

```
Usage:
cmsrel CMSSW_3_X_Y
cd CMSSW_3_X_Y/src
cmsenv
addpkg L1Trigger/Configuration
cd L1Trigger/Configuration/test
./L1EmulatorFromRaw.sh Global_Tag EventSampleType

Global tag must be given in one of the following formats:
auto:mc
auto:startup
auto:com10
MC_3XY_V22
START3X_V22
GR10_P_V3

Default: auto:com10

Event sample type follow the cmsDriver convention:
data
mc
empty string default to mc
```

Running L1EmulatorFromRaw.sh without arguments, or with one of the arguments `--help` or `-help` will print the actual recipe.

Customization python script

The customization python script `customise_11EmulatorFromRaw.py` used to modify the `cmsDriver` for the steps "RAW2DIGI,L1" calls the following modules from `L1Trigger_custom.py`:

- a module to (re-)run the L1 emulator starting from a RAW file [\[link\]](#)
- a module to change any or more of L1 menu, prescale factors, trigger masks for both algorithm and technical triggers [\[link\]](#) (optional). By defaults, no changes are done to the values obtained from the global tag
- a module to customize the output commands, starting from the event content defined in the `cmsDriver` command (optional). By default, no changes are done to event content
- a module to print the L1 trigger report for the new L1 GT readout record (optional)

A user does not need to do any changes in `customise_11EmulatorFromRaw.py`. Changes, if desired or needed, should be done in the corresponding module definitions from `L1Trigger_custom.py`.

Run with a different configuration

Run with the L1 trigger configuration from a different global tag

To run with a different L1 trigger configuration, one can use a different global tag having the desired L1 trigger configuration; the new global tag must however be compatible with the release used to re-run the L1 trigger emulators, see [Global Tags for Conditions Data](#) wiki page

Run with the same global tag, but overwrite some of the L1 trigger records

The customization fragment `customise_11TriggerConfiguration.py` allows to change, via boolean flags, the following L1 trigger conditions

- reset algorithm trigger masks
- reset technical trigger masks
- unprescale algorithm triggers (all prescale factors set to 1)
- unprescale technical triggers (all prescale factors set to 1)
- overwrite the L1 trigger menu

By default, all the above flags are set to `False`.

In order to change the L1 menu, in addition to turn on the logical flag from `customise_11TriggerConfiguration.py`, one has to set the new menu in the customization block `def customiseL1Menu(process)` : from `L1Trigger_custom.py`. The following alternatives are available

- menu from a global tag
- menu from an SQL file
- menu from an XML files

More options will be added eventually.

Note: the customization code for overwriting a L1 menu does not check if the new menu is consistent with the global tag used. It is the responsibility of the user to ensure this consistency.

Overwrite the L1 menu with a menu from an SQL file

In order to overwrite the L1 menu with a menu from an SQL file, one has to provide the SQL file in a location where it can be read by the job and then modify in the block `customiseL1Menu` from `L1Trigger_custom.py`

("user choices") the location to the actual one.

Overwrite the L1 menu with a menu from an XML files

In order to overwrite the L1 menu with a menu from an XML file, the file must be copied in one of the directories from `[[][]]`, then modify in the block `customiseL1Menu` from `L1Trigger_custom.py` ("user choices") the location to the actual one (both directory name `luminosityDirectory` and file name `useXmlFile`).

Customization of output commands

By default, the event content save will be the event content given in the `cmsDriver` command, via the `--eventcontent` flag. The event content resulted from the event content flag can be modified in the `def customiseOutputCommands (process) :` block from `L1Trigger_custom.py`. Some examples (commented) are available in the existing block.

Print the L1 trigger report

It is recommended to print the L1 trigger report when running the script `L1EmulatorFromRaw.sh` (set to `True` by default). The configuration of the report can be changed in the block `def customiseL1TriggerReport (process) :` from `L1Trigger_custom.py`.

Obsolete: L1 trigger configured via dummy producers

In previous releases, the L1 configuration was produced by dummy producers (via `L1DummyConfig_cff.py`). So you can change the value of any L1 configuration parameter using standard python assignment. This configuration style is not supported any more.

Workflow 2 : Re-running the L1 GT emulator from raw data

This allows the user to re-run the L1 GT emulator with a different menu, prescale factors and trigger masks then used for the original production.

An example can be found at `L1Trigger/Configuration/test/L1GtEmulatorFromRaw.sh`.

All the considerations applying to `L1EmulatorFromRaw.sh` apply also to `L1GtEmulatorFromRaw.sh`.

Running L1 trigger emulator or L1 GT emulator and HLT

If HLT is to be re-run as well, the HLT needs to be configured to use the newly generated L1 digis, instead of those obtained from the raw data. The TSG group is (in task of) providing the corresponding recipes, please read the TSG wiki.

Review status

Reviewer/Editor and Date	Comments
JimBrooke - 18 Feb 2008	Recipe for 2XX releases
VasileGhete - 02-Jun-2010	Updated to 3XX releases

Responsible: Vasile Ghete

This topic: CMSPublic > SWGuideReRunningL1

Topic revision: r17 - 2012-02-10 - ArnoHeister



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)